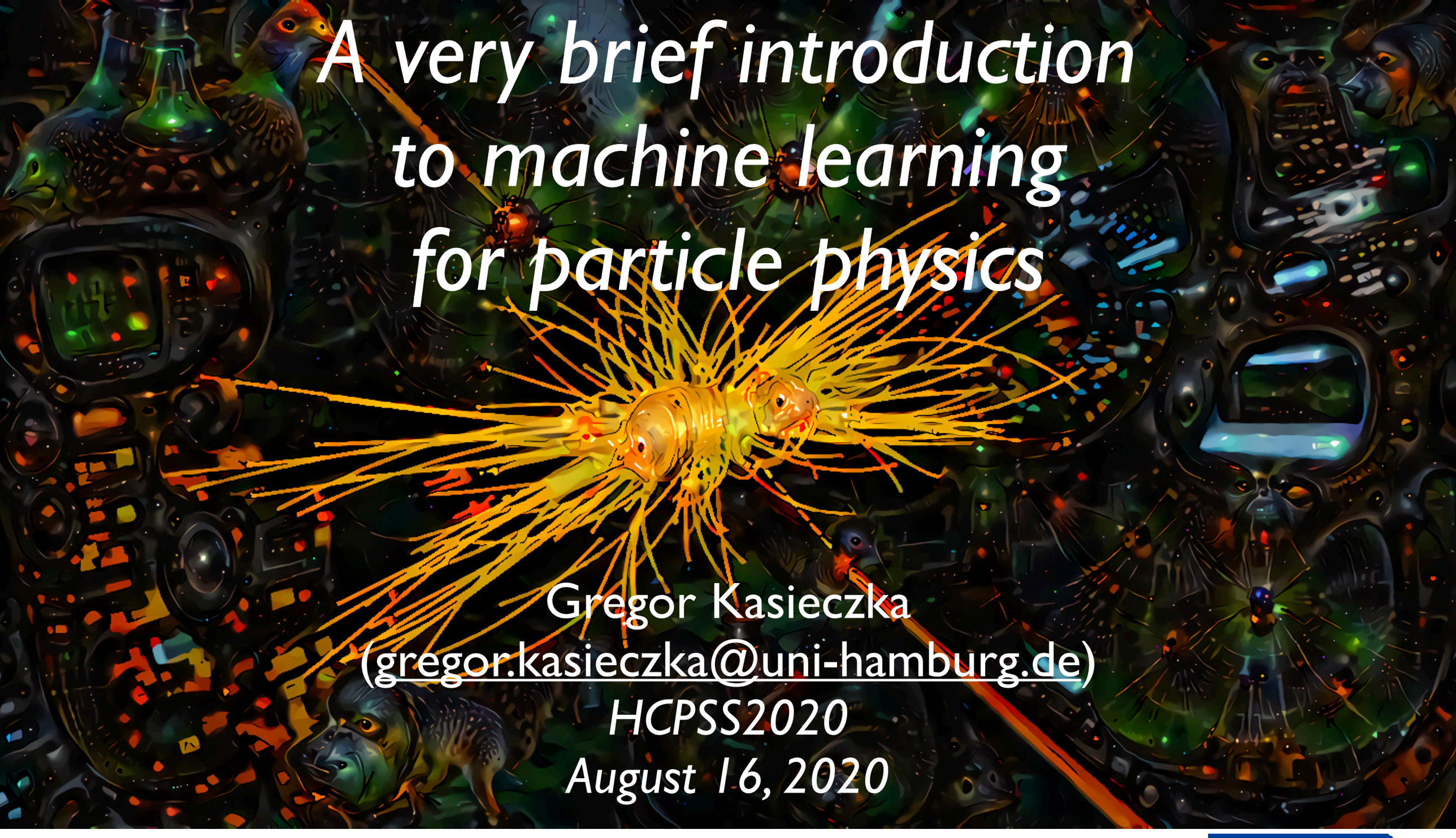


A very brief introduction to machine learning for particle physics

Gregor Kasieczka
(gregor.kasieczka@uni-hamburg.de)

HCPSS2020
August 16, 2020

CLUSTER OF EXCELLENCE
QUANTUM UNIVERSE

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Emmy Noether-Programm
Deutsche Forschungsgemeinschaft DFG

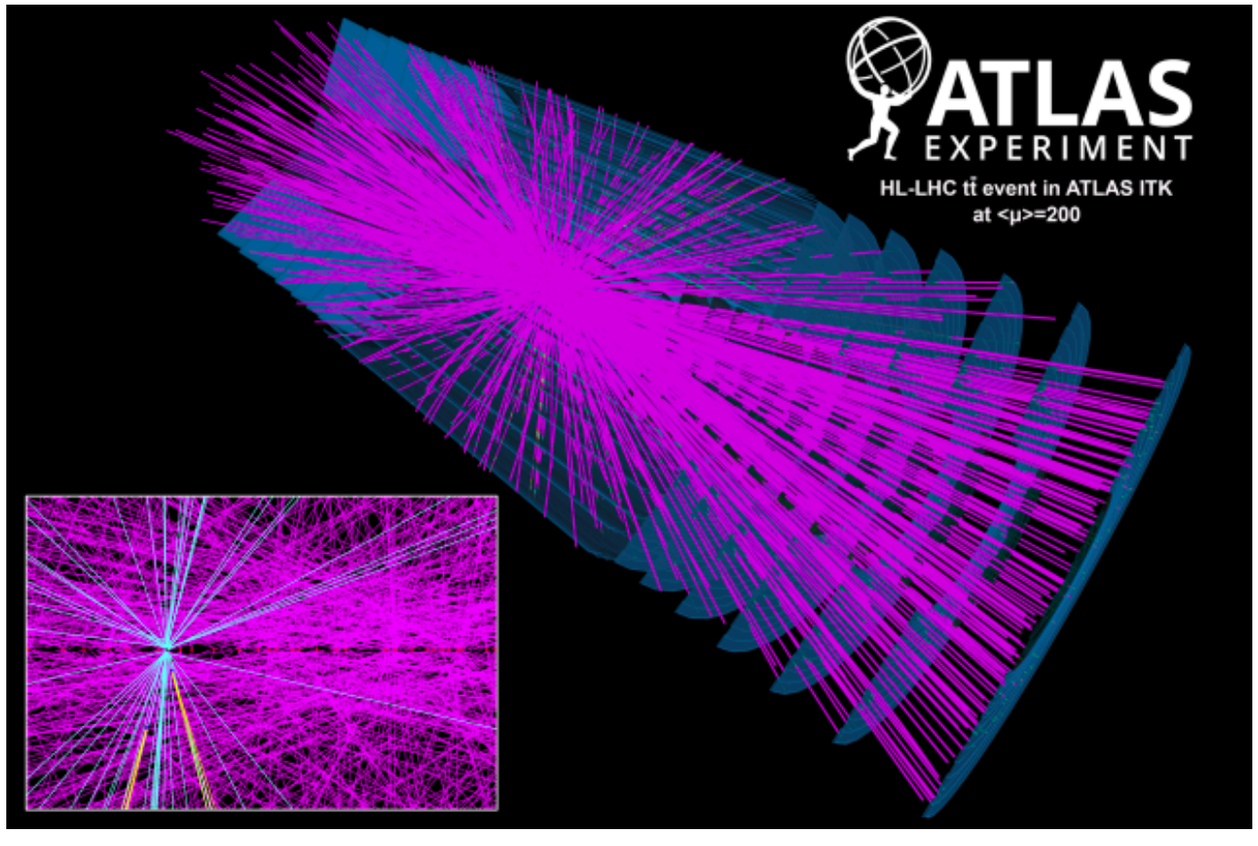
PIER
Partnership of Universität Hamburg and DESY

Bundesministerium für Bildung und Forschung

Why?

- Precision measurements and searches for new physics need
  - better tools to identify and measure particles and processes
  - higher accuracy and speed
- Finding unknown signatures and measurements need
  - new ways of analysing data
- Future data taking with higher collision rates needs:
  - faster reconstruction and triggering
  - faster event generation and detector simulation

(a) promising answer: **Deep Learning**

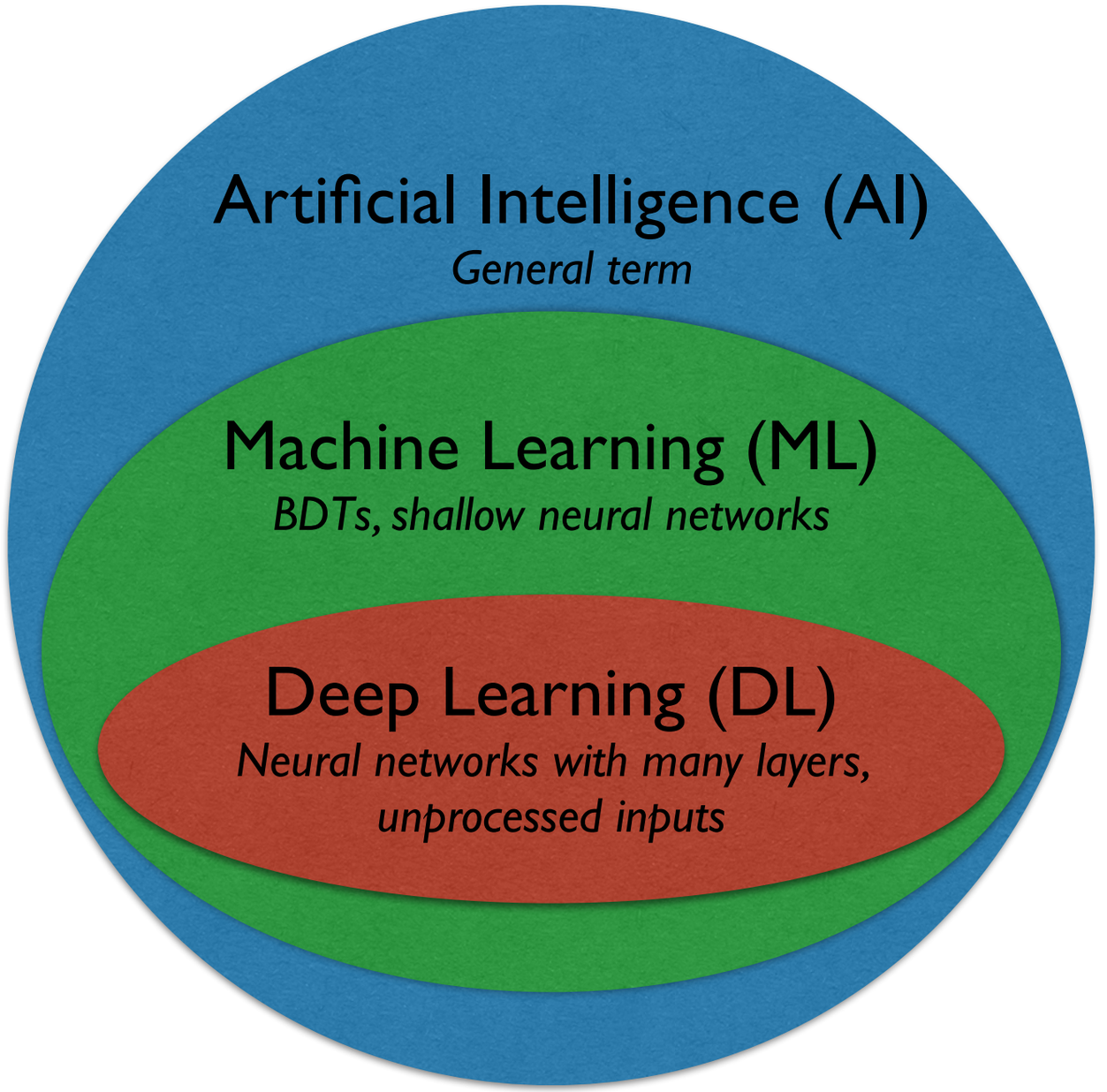


Program

- The basics
- Supervised particle tagging & architectures
- Generative models
- Unsupervised searches
- Some final words

The basics

Terminology

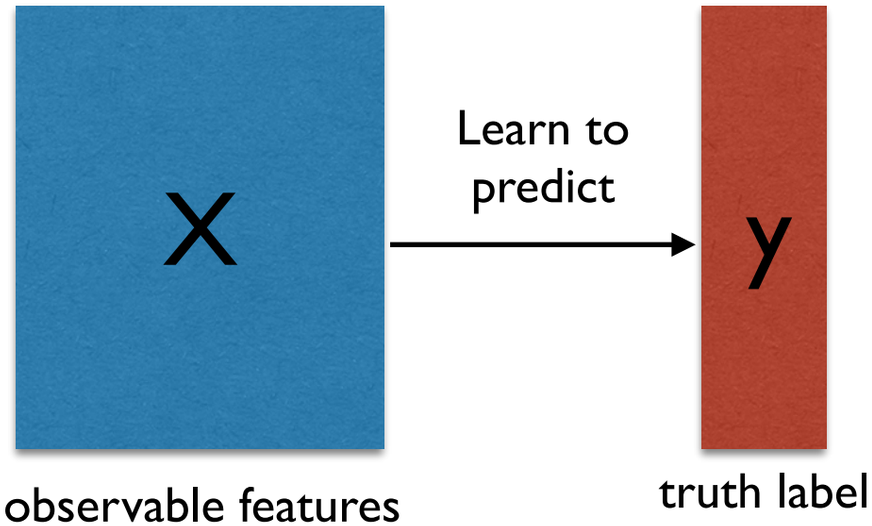


Tasks

Supervised Learning

Attempt to infer some target *(truth label)*: classification (*jet flavour tagging*) or regression (*energy calibration*)

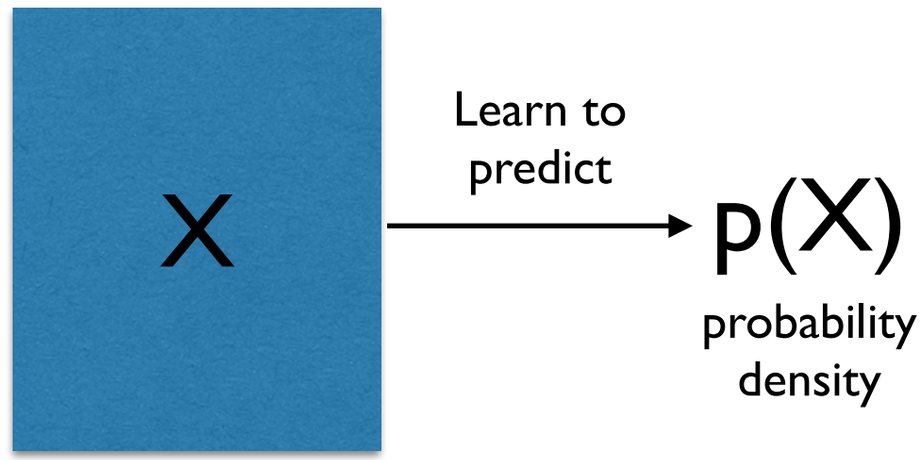
Need training dataset with known labels (typically from MC simulation)



Unsupervised

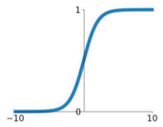
No target, learn the probability distribution

Useful for generative models and anomaly detection.

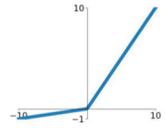


Activation Functions

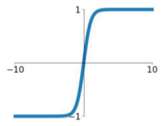
Sigmoid
$\sigma(x) = \frac{1}{1+e^{-x}}$



Leaky ReLU
$\max(0.1x, x)$

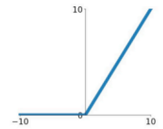


tanh
$\tanh(x)$

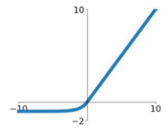


Maxout
$\max(w_1^T x + b_1, w_2^T x + b_2)$

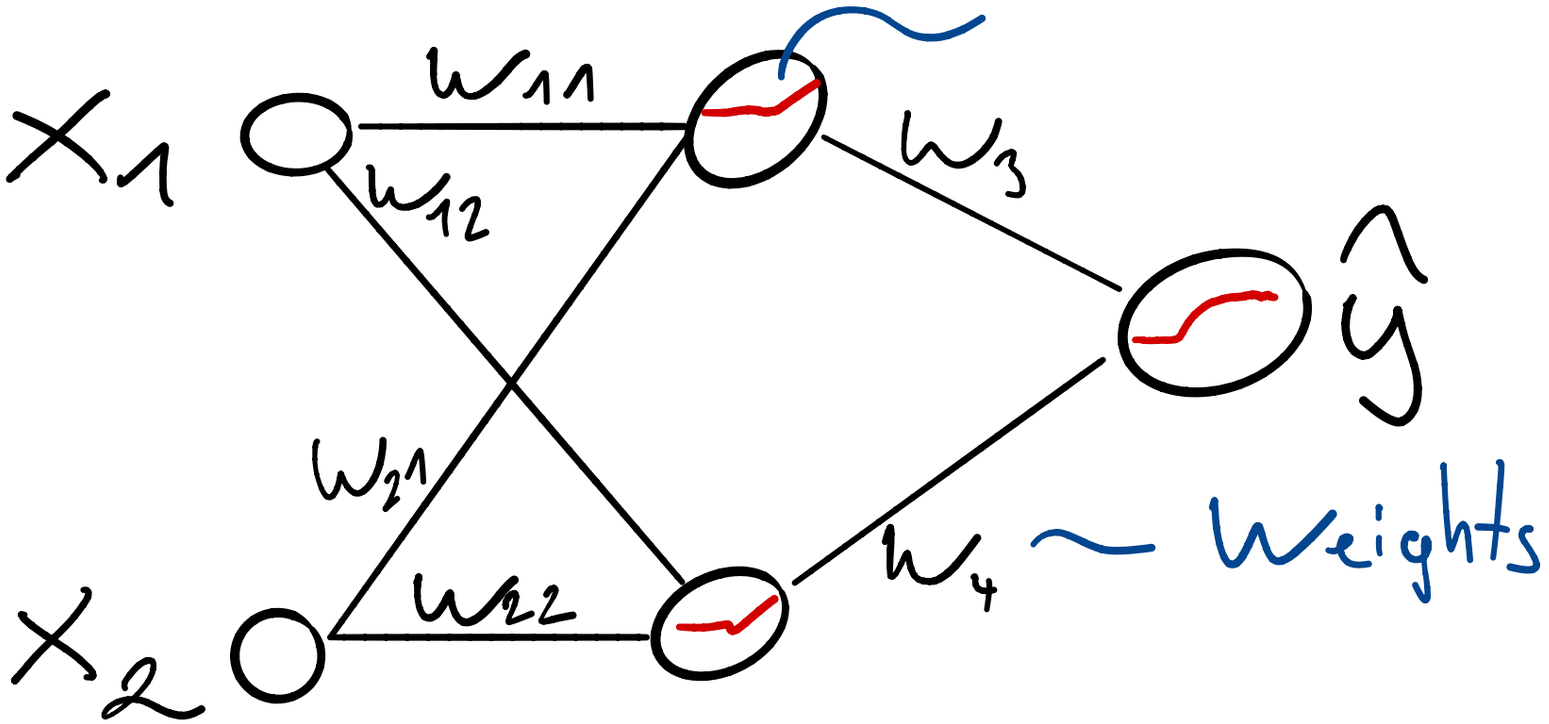
ReLU
$\max(0, x)$



ELU
$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$



Activation Function



w_i ~ Weights

Input Layer

Hidden Layer

Output Layer

How do networks learn?

- *Backpropagation + Gradient descent*
- Pass input (x_1, x_2, …) to networks
- From output ($\hat{y}$) and true value (y) calculate optimisation target (*loss function L*)
- For example: Mean Squared Error (MSE) for regression:

$$L(y, \hat{y}) = (y - \hat{y})^2$$

- Find gradient of loss function with respect to weights
- Use gradient to find new weights

$$w_{t+1} = w_t - \eta \frac{\partial L}{\partial w_t} \equiv w_t - \eta \nabla L(w_t)$$

Learning rate

- Practically, this is taken care of by an optimiser algorithm *(e.g. Adam as default)*

Classification Loss

- Classification loss function: **Cross entropy** between true labels (p) and network output (q): $H(p,q) = -\sum p_i \ln q_i$

- Rewrite as: $-\sum p_i \ln q_i = H(p) + D_{KL}(p||q)$

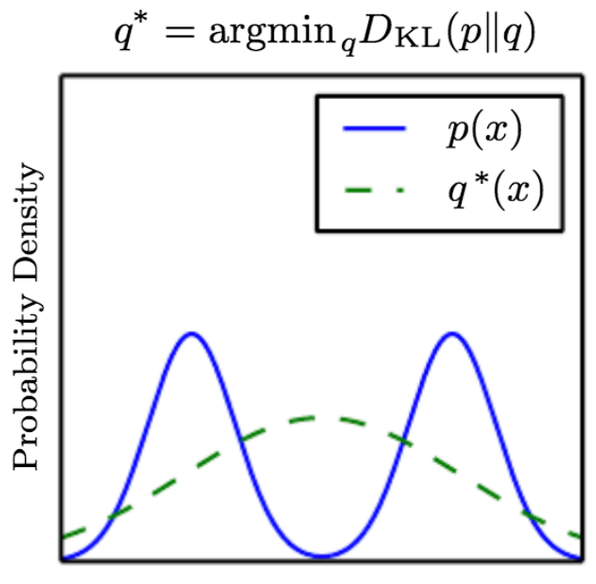
$$H(p) = -\sum p_i \ln p_i$$

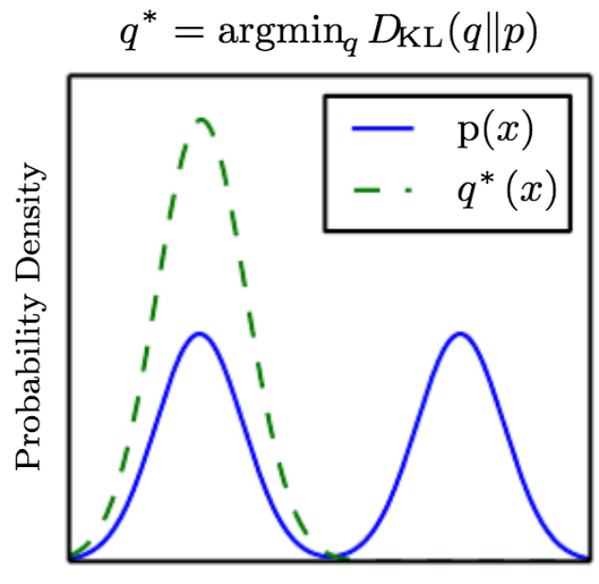
- Entropy: *Average amount of information produced by measurements of random variable (Notice similarity to Gibbs entropy)*

For fixed true labels, the cross entropy measures the differences bewegen truth and network prediction

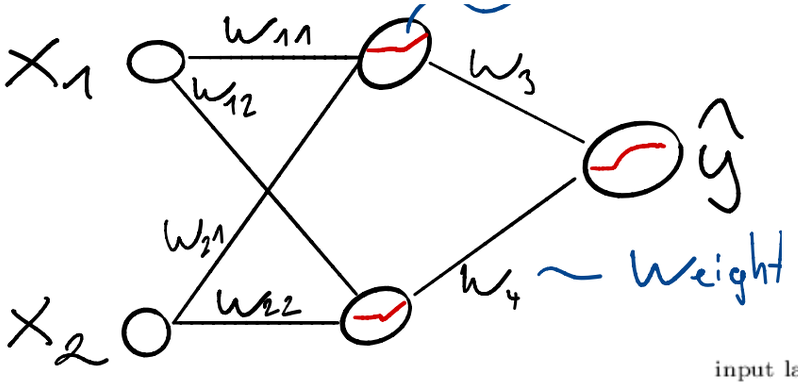
Kullback-Leibler Divergence:
(measure of difference between two distributions)

$$D_{\mathrm{KL}}(p||q) = -\sum p(i) \log \frac{q(i)}{p(i)}$$

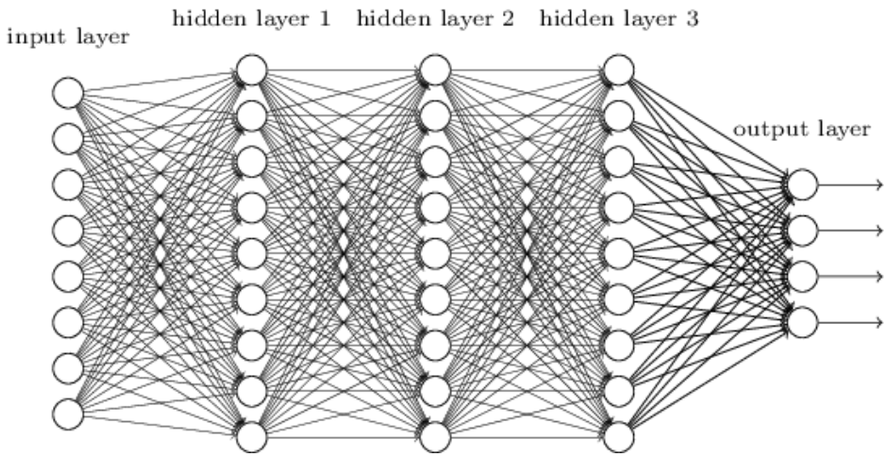




Complexity



6 weights



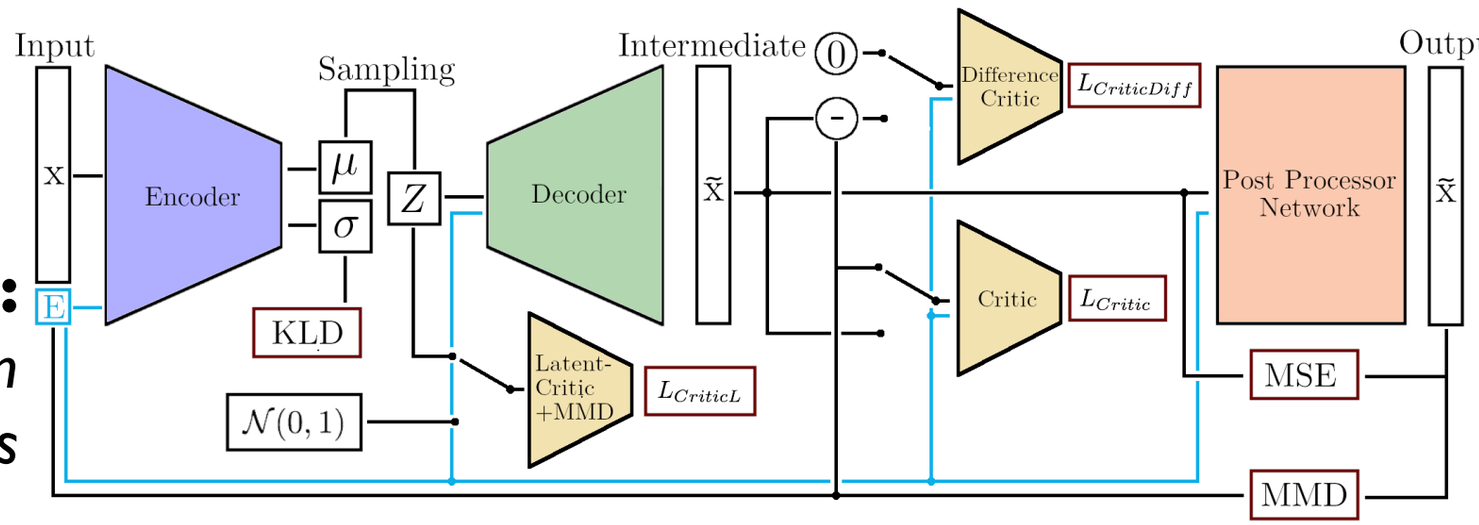
300 weights

stage	output	ResNet-50	**ResNeXt-50 (32×4d)**
conv1	112×112	7×7, 64, stride 2	7×7, 64, stride 2
conv2	56×56	3×3 max pool, stride 2	3×3 max pool, stride 2
		[1×1, 64 3×3, 64 1×1, 256] ×3	[1×1, 128 3×3, 128, C=32 1×1, 256] ×3
conv3	28×28	[1×1, 128 3×3, 128 1×1, 512] ×4	[1×1, 256 3×3, 256, C=32 1×1, 512] ×4
conv4	14×14	[1×1, 256 3×3, 256 1×1, 1024] ×6	[1×1, 512 3×3, 512, C=32 1×1, 1024] ×6
conv5	7×7	[1×1, 512 3×3, 512 1×1, 2048] ×3	[1×1, 1024 3×3, 1024, C=32 1×1, 2048] ×3
	1×1	global average pool 1000-d fc, softmax	global average pool 1000-d fc, softmax
# params.		**25.5**×10^6	**25.0**×10^6
FLOPs		**4.1**×10^9	**4.2**×10^9

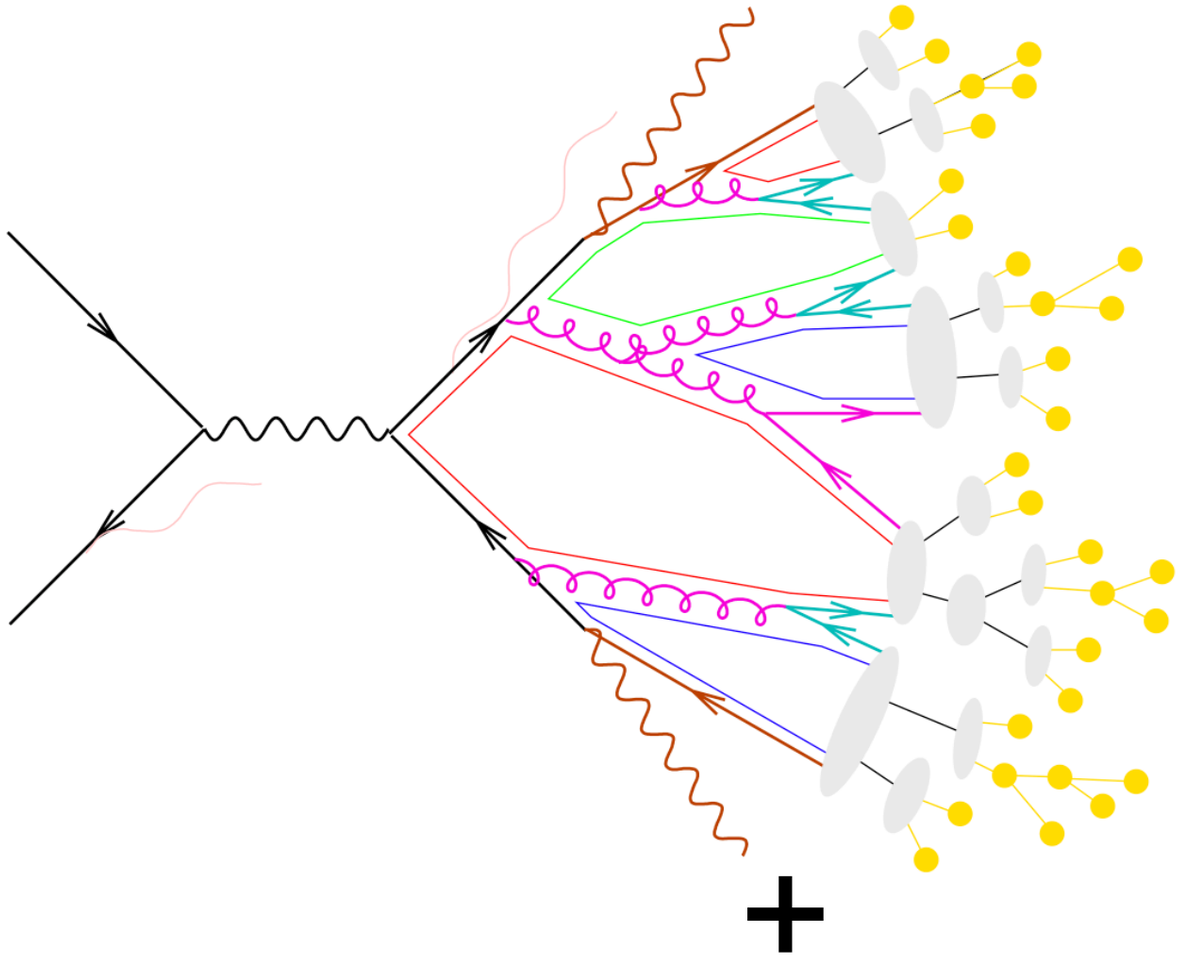
25 million weights:
2016 state of the art for image classification

Deep Learning:
Complex network + low level inputs

71 million weights:
2020 generative network from physics

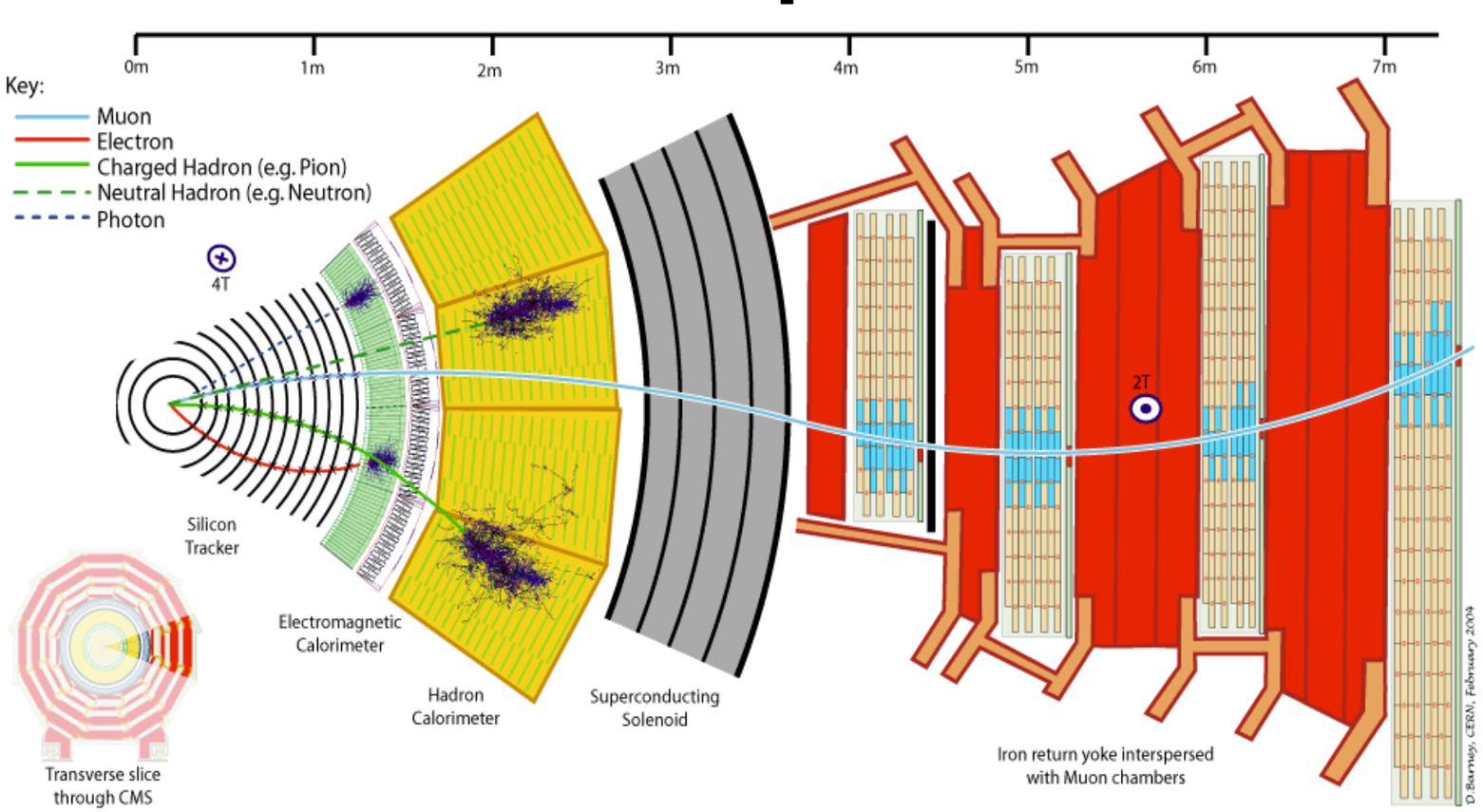


Supervised particle tagging & architectures



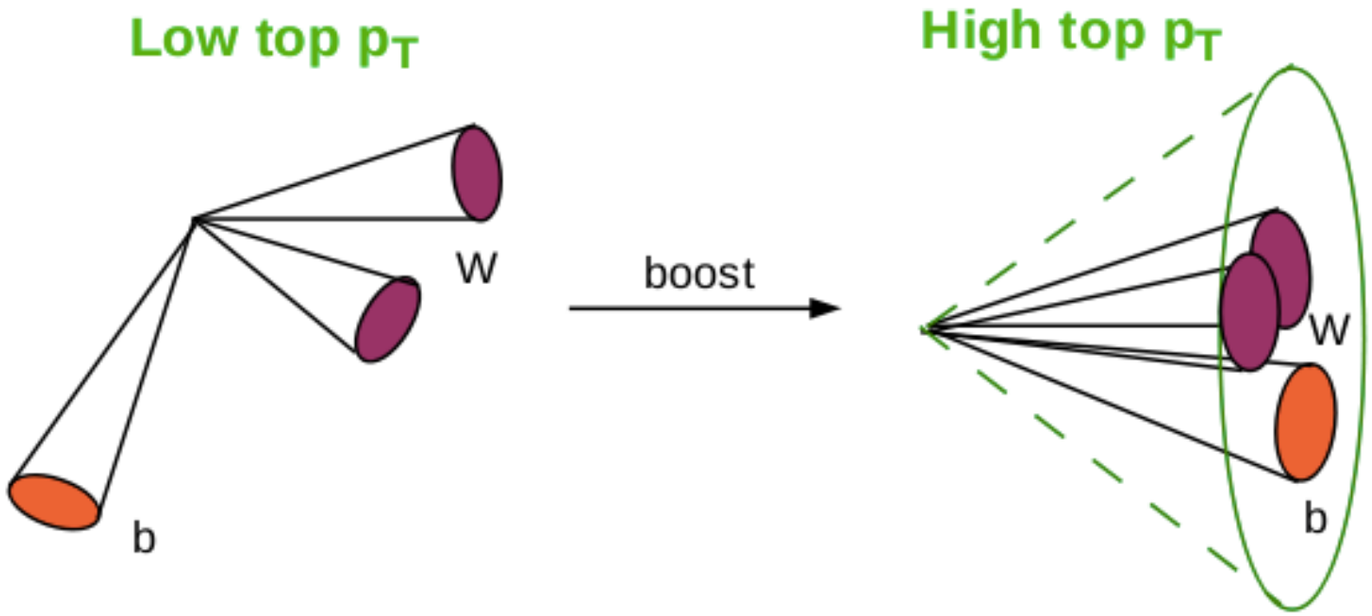
- hard scattering
- (QED) initial/final state radiation
- partonic decays, e.g. $t \to bW$
- parton shower evolution
- nonperturbative gluon splitting
- colour singlets
- colourless clusters
- cluster fission
- cluster → hadrons
- hadronic decays

We want to infer underlying physics from measurements in the detector.



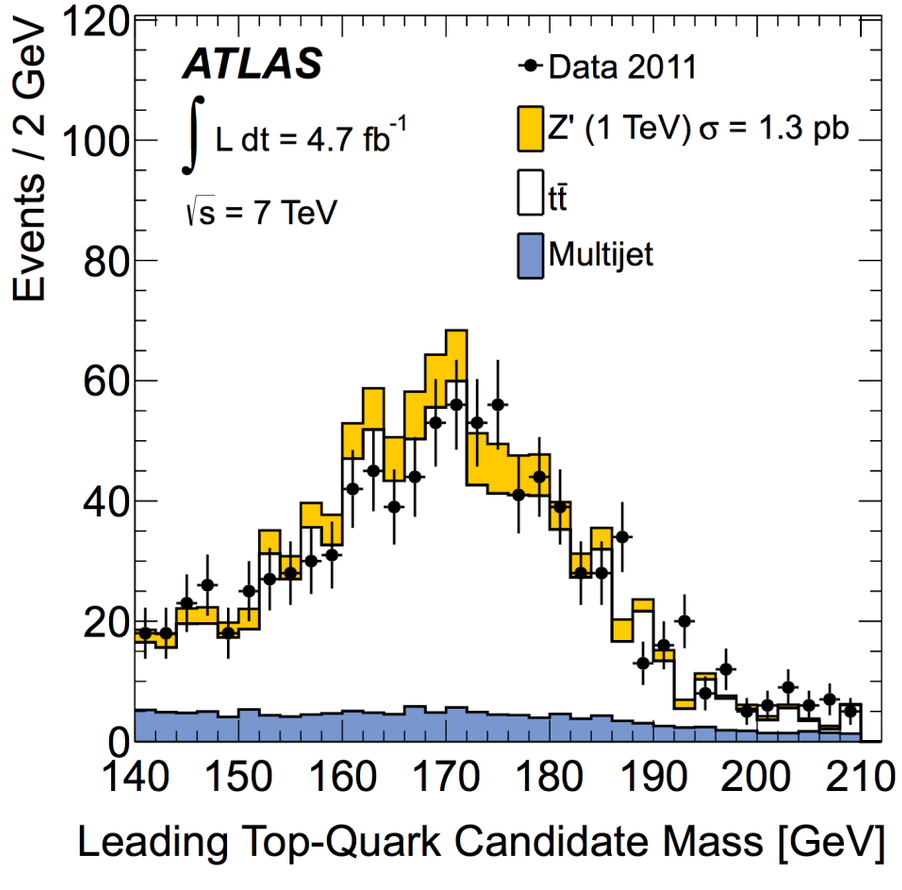
How can deep neural networks assist us?

Heavy Resonance Tagging



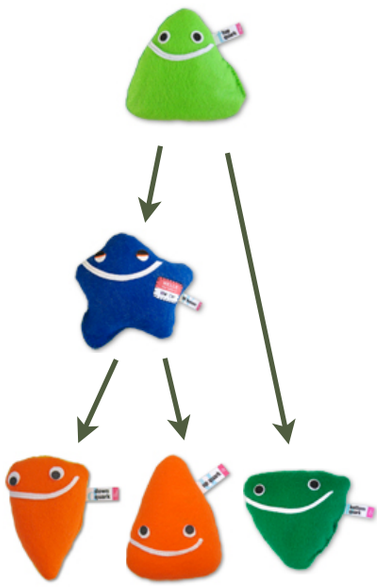
- Hadronically decaying top/Higgs/W/Z
- Contained in one (large-R) jet
- How to distinguish from light quark/gluon jets (and from each other)
- For new physics searches (and SM studies)

- Mass
  Calculate using a grooming algorithm (eg mMDT/softdrop or pruning)
- Centers of hard radiation
  n-subjettiness or energy correlation functions
- Flavour
  b tagging of large-R jets or subjets
- Combinations

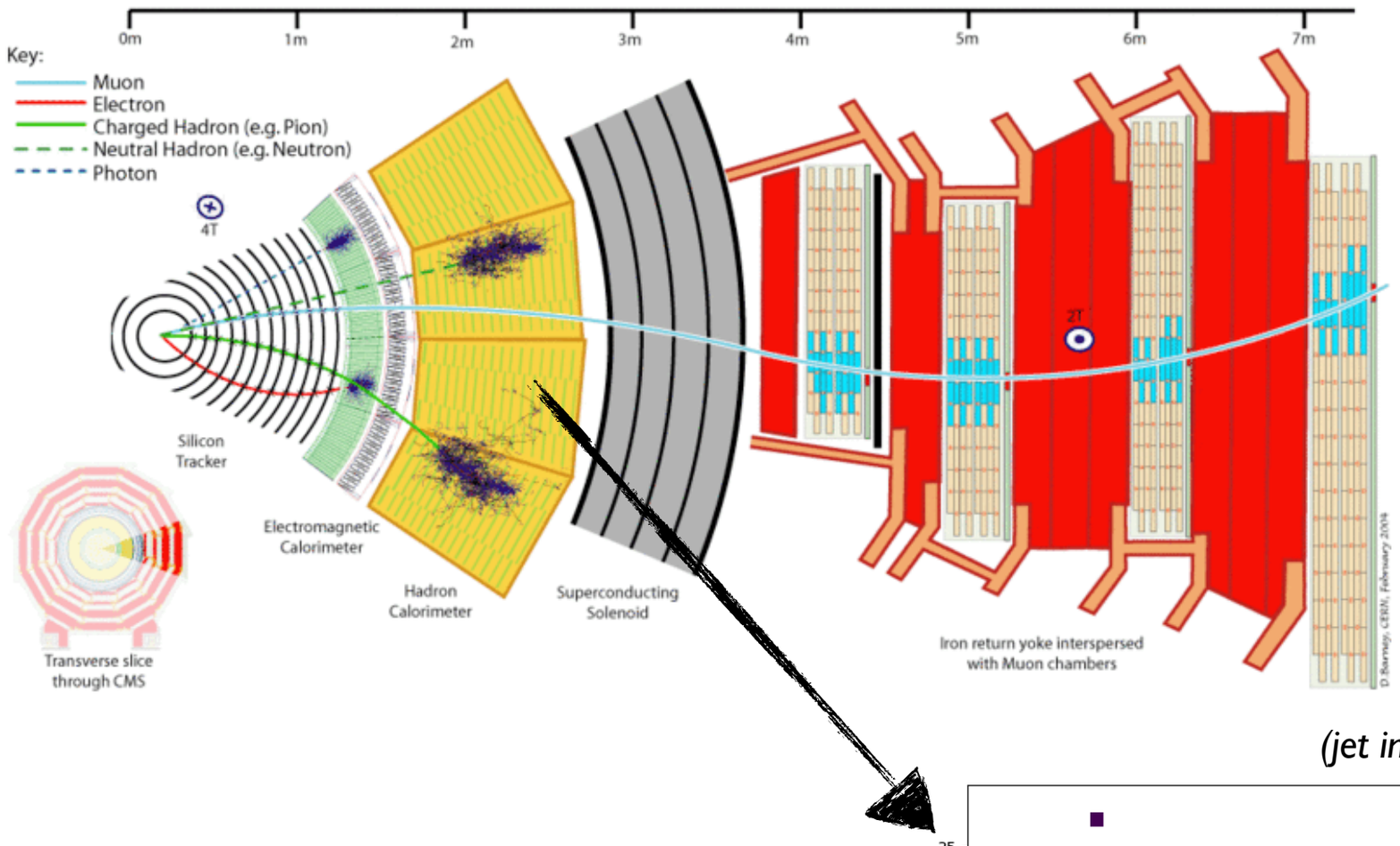


Towards an Understanding of the Correlations in Jet Substructure
D Adams et al (BOOST 2013 Participants), Eur.Phys.J. C75
Top Tagging, T Plehn, M Spannowksy, J.Phys. G39 (2012) 083001
Boosted Top Tagging Method Overview, GK, Proc. Top2017

Top Quark



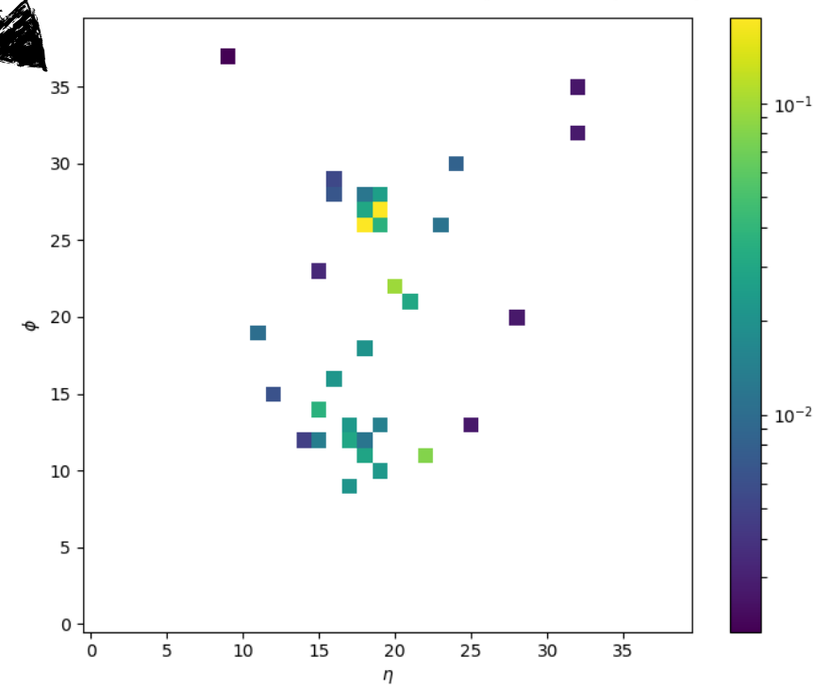
+



(jet images by C Daza)

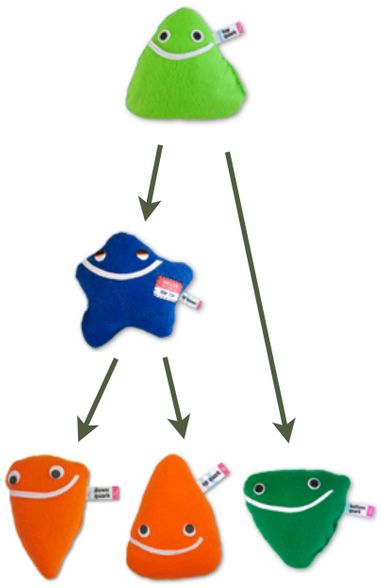
- *Measure particle energies in calorimeter*
- *Reconstruct jet from individual measurements*
- *Image preprocessing*
  - *center, rotate, mirror, pixelate, trim, normalise*

=

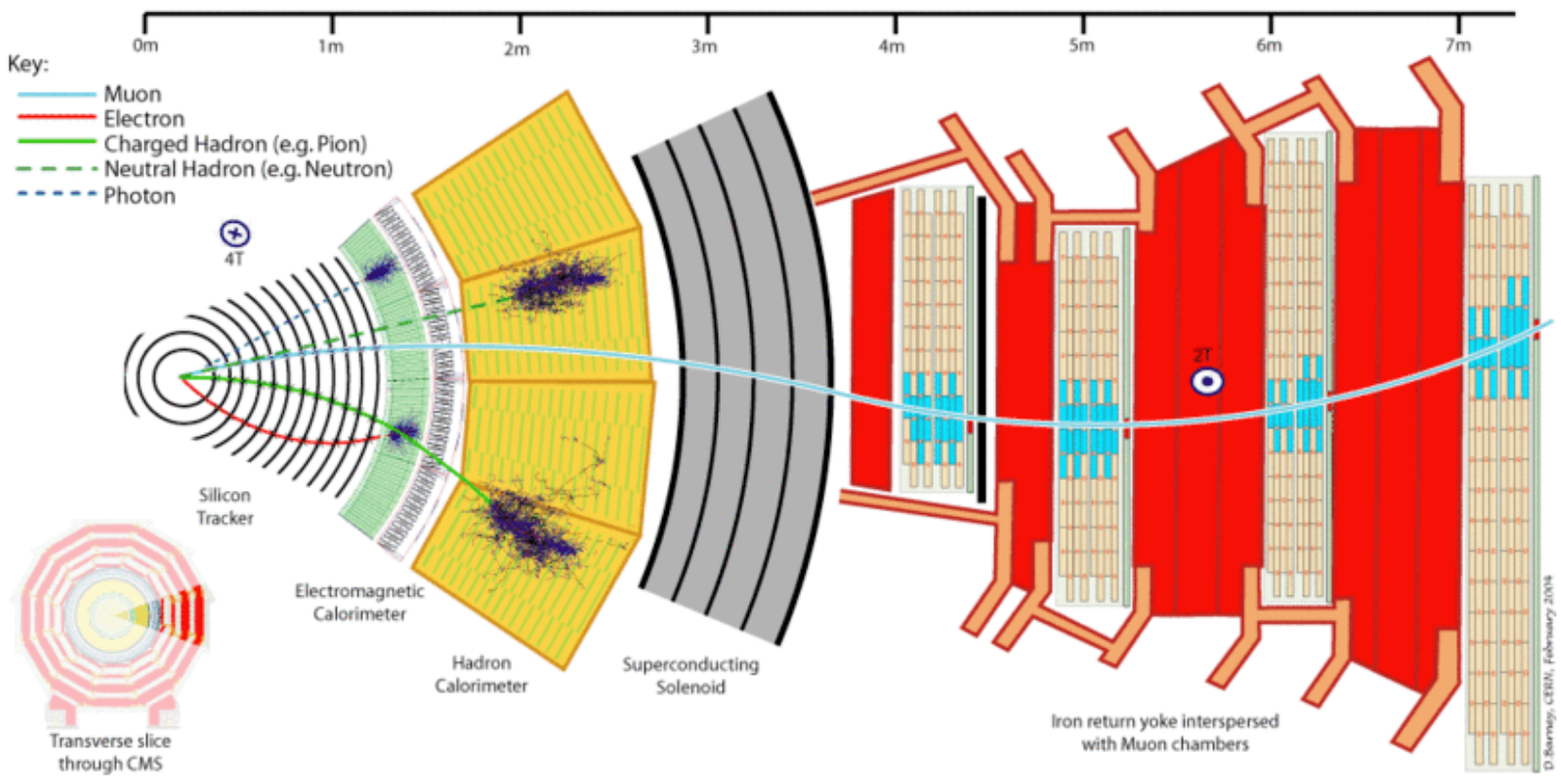


1 image

Top Quark



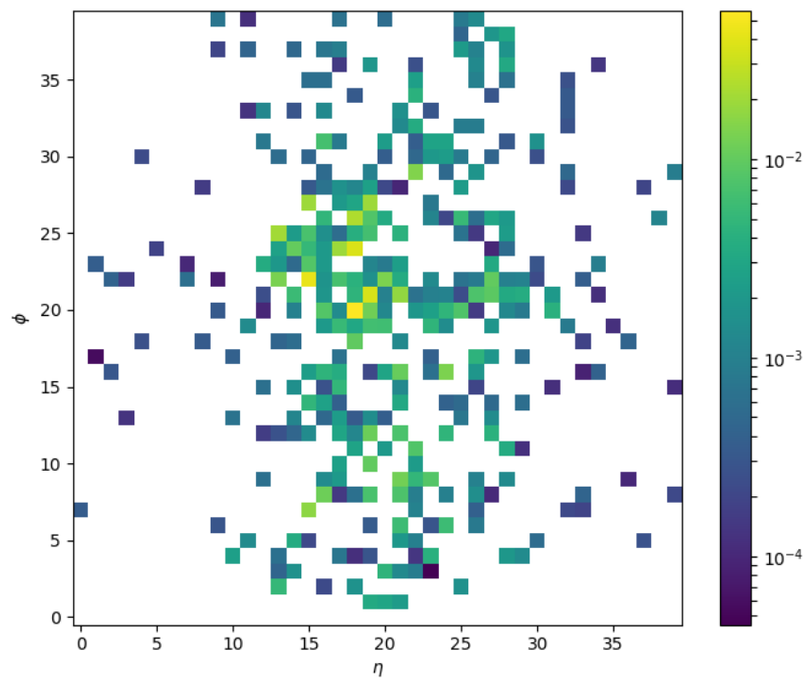
+



(jet images by C Daza)

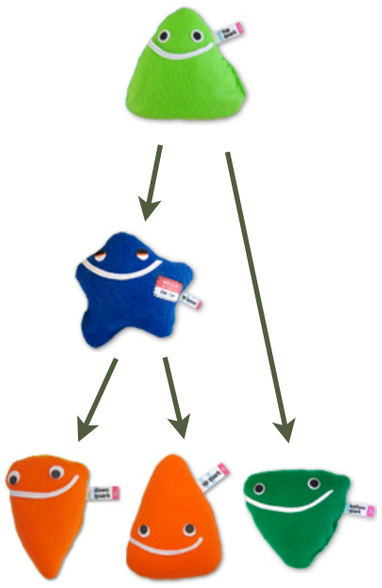
- *Measure particle energies in calorimeter*
- *Reconstruct jet from individual measurements*
- *Image preprocessing*
  - *center, rotate, mirror, pixelate, trim, normalise*

=

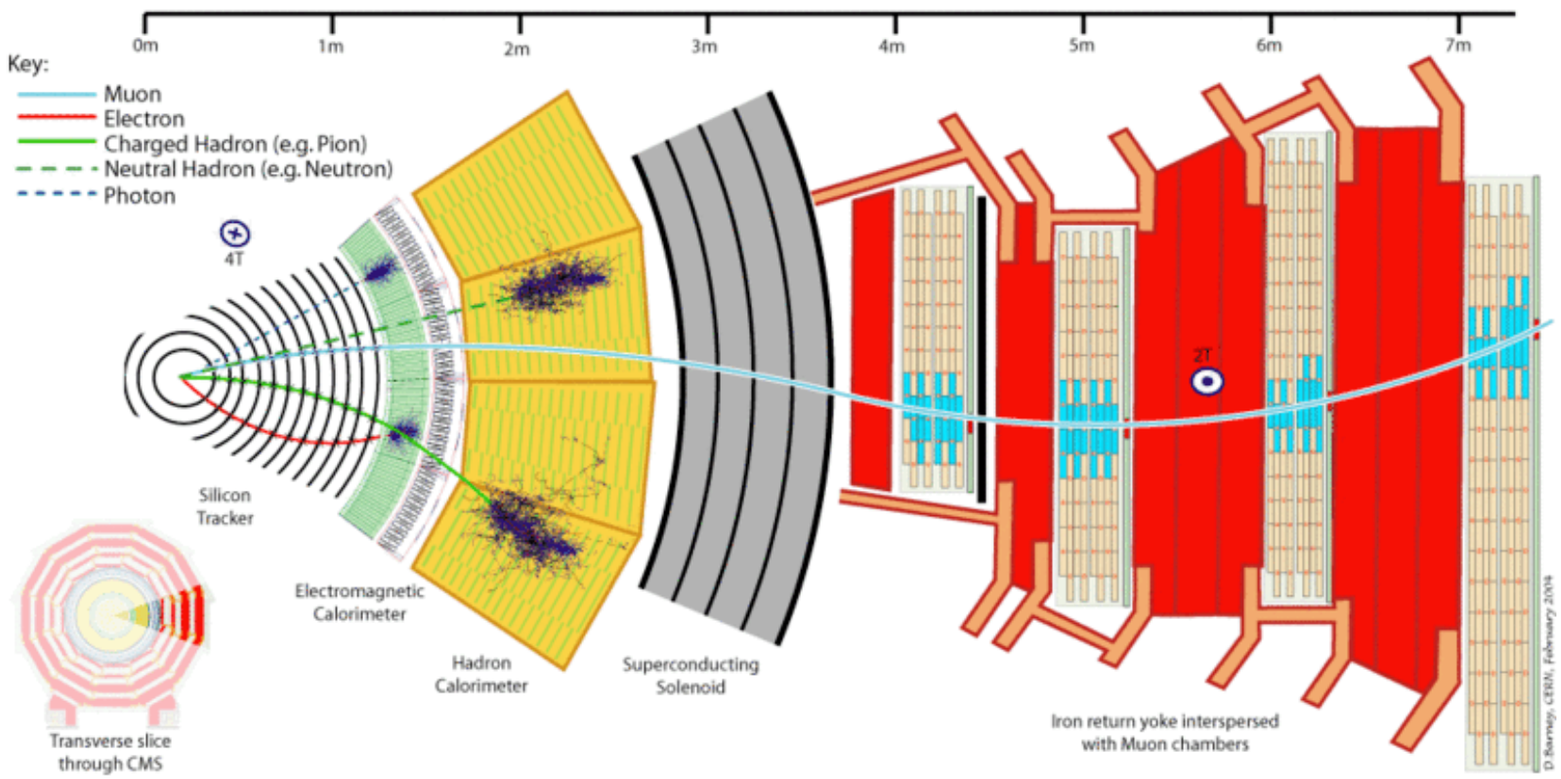


10 image average

Top Quark



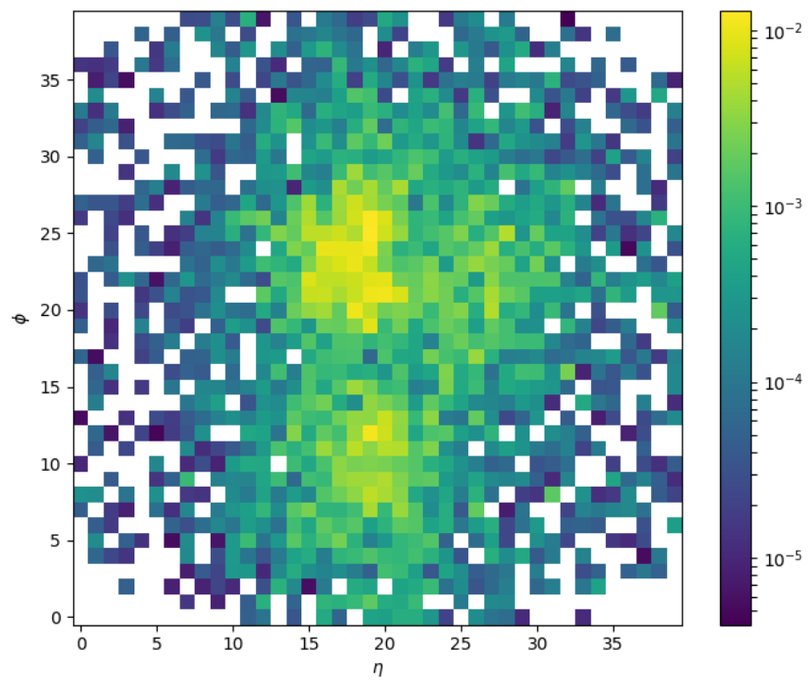
+



(jet images by C Daza)

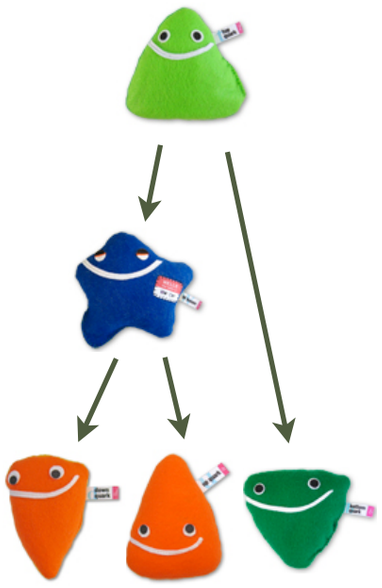
- *Measure particle energies in calorimeter*
- *Reconstruct jet from individual measurements*
- *Image preprocessing*
  - *center, rotate, mirror, pixelate, trim, normalise*

=

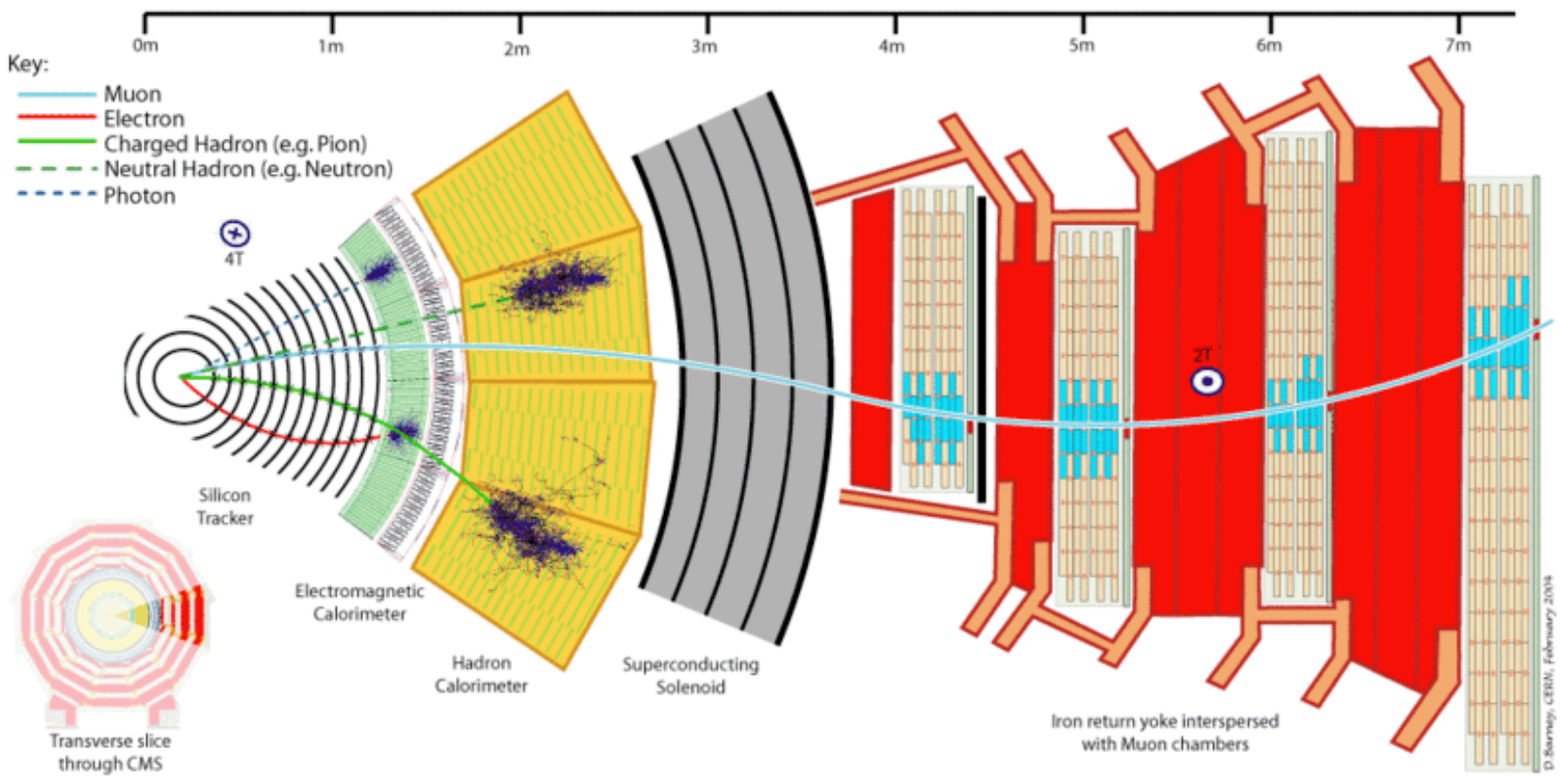


100 image average

Top Quark



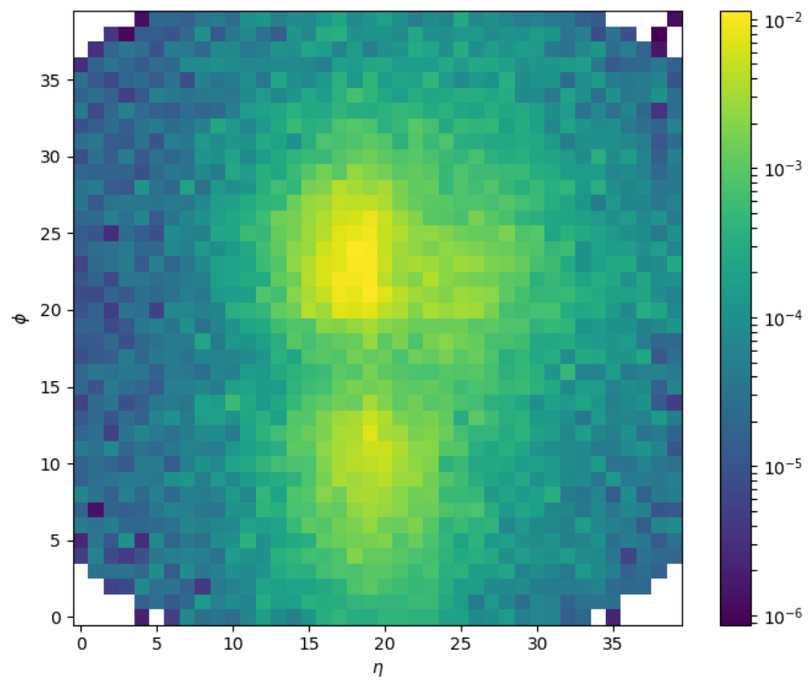
+



(jet images by C Daza)

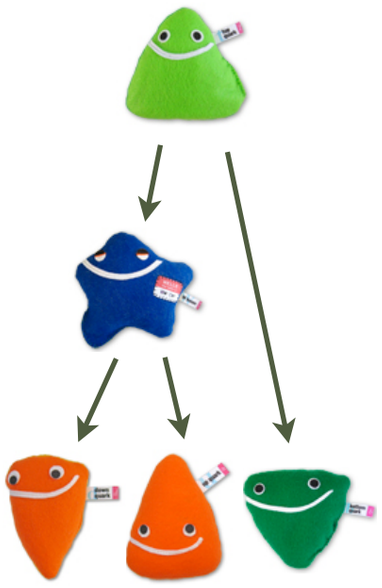
- *Measure particle energies in calorimeter*
- *Reconstruct jet from individual measurements*
- *Image preprocessing*
  - *center, rotate, mirror, pixelate, trim, normalise*

=

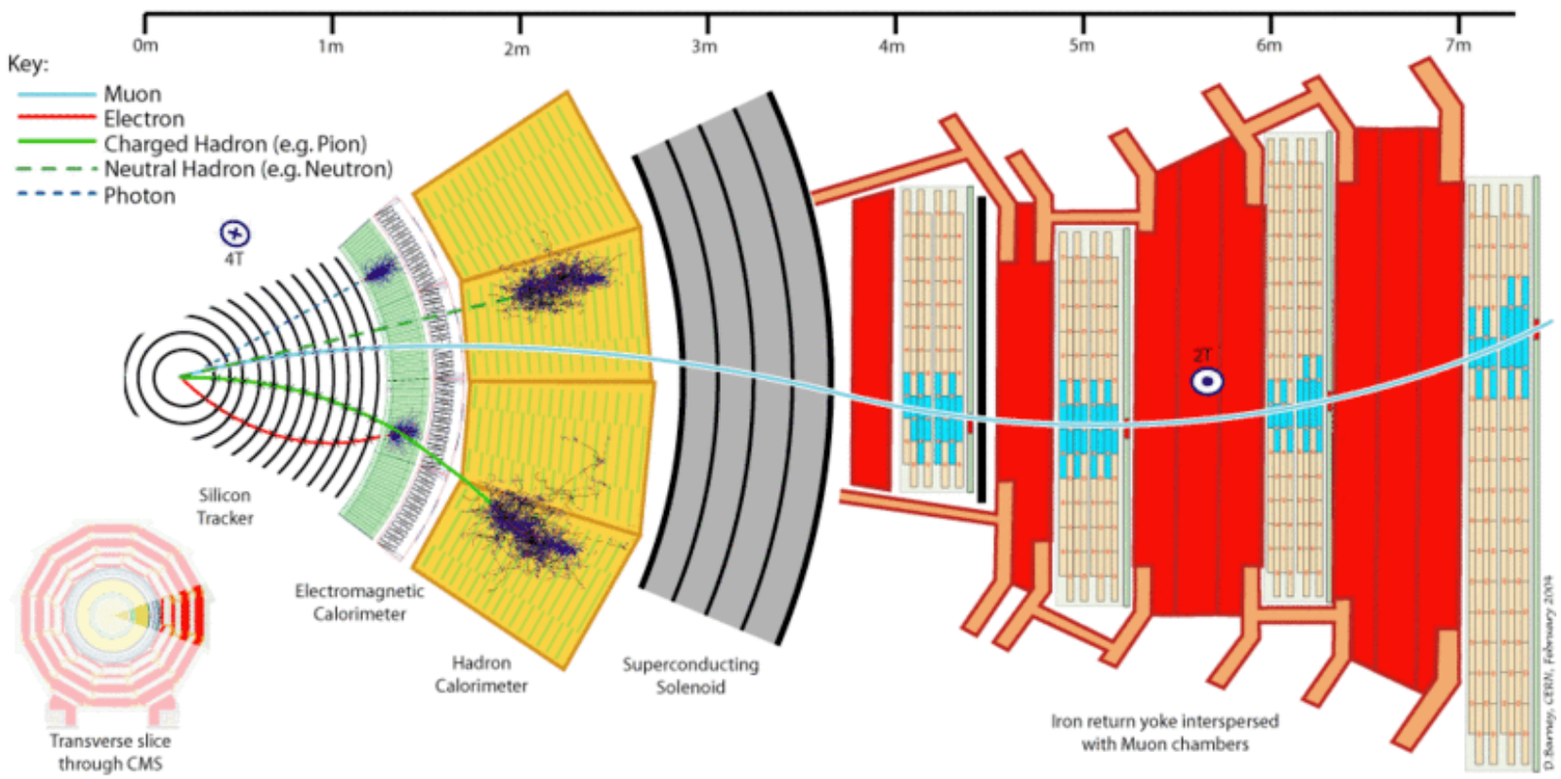


1000 image average

Top Quark



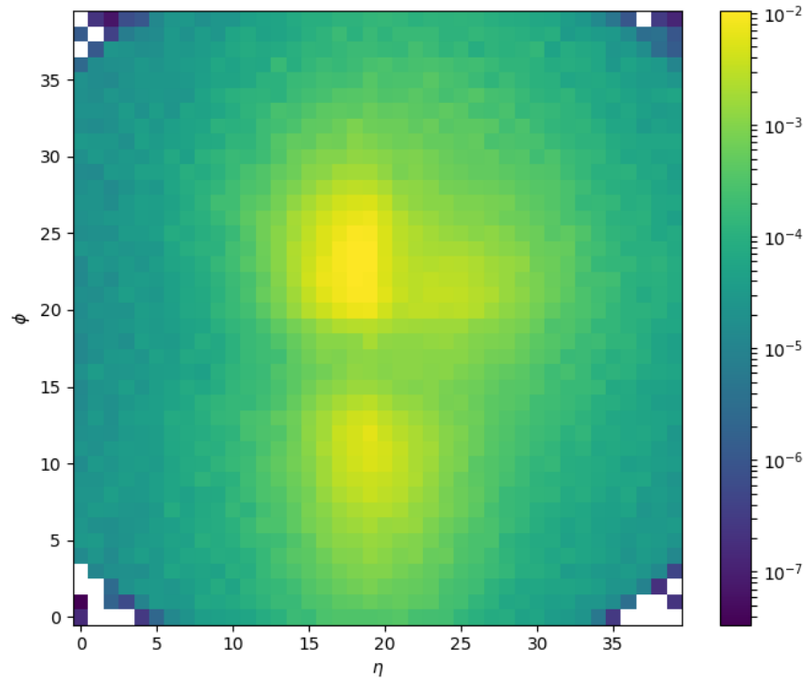
+



(jet images by C Daza)

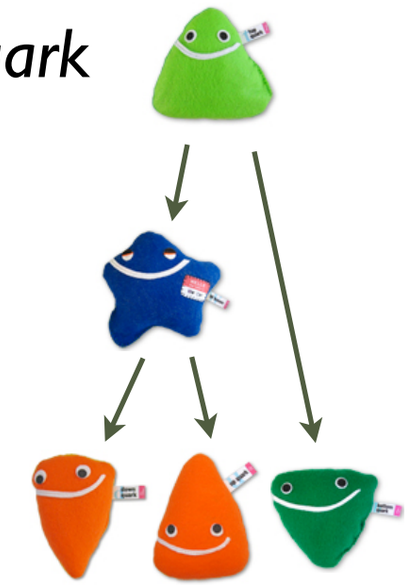
- *Measure particle energies in calorimeter*
- *Reconstruct jet from individual measurements*
- *Image preprocessing*
  - *center, rotate, mirror, pixelate, trim, normalise*

=

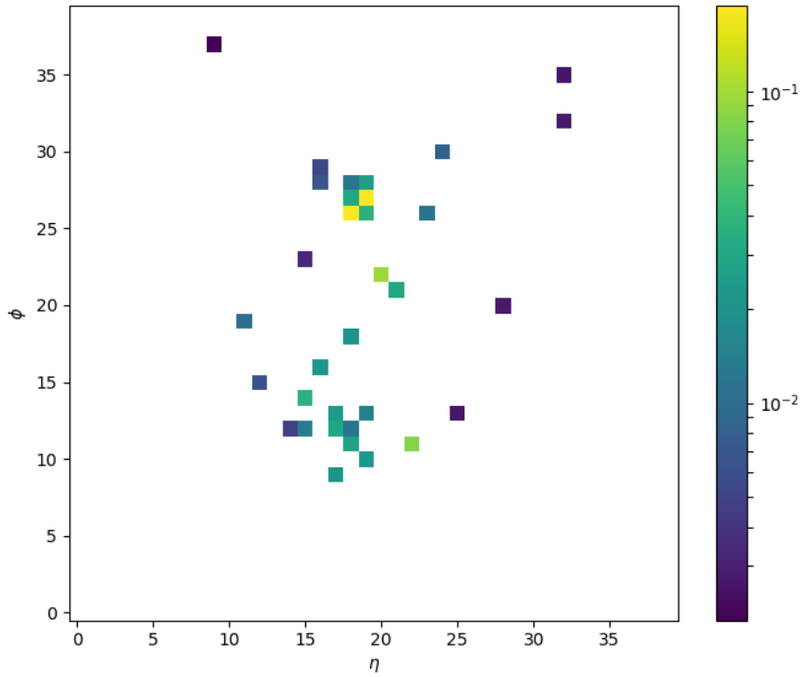


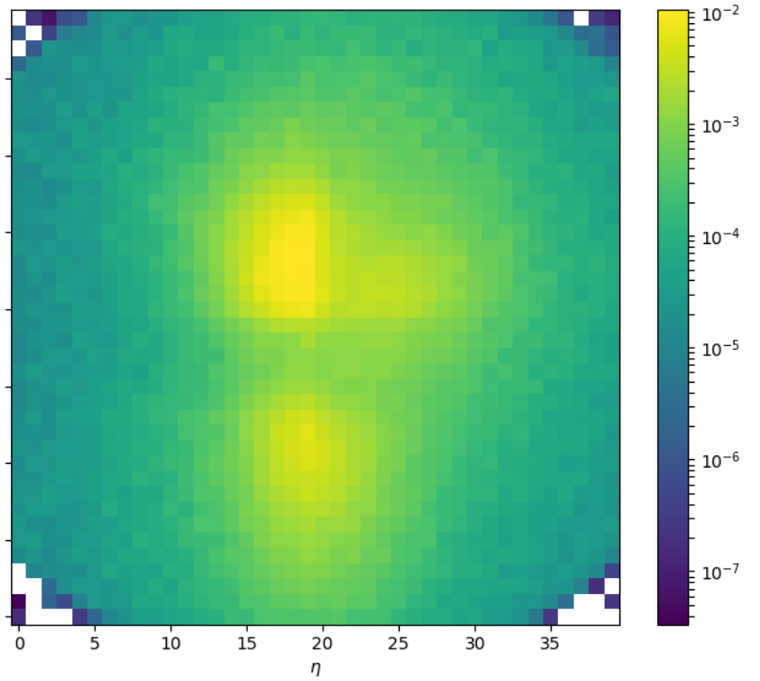
10000 image average

Top Quark Jet



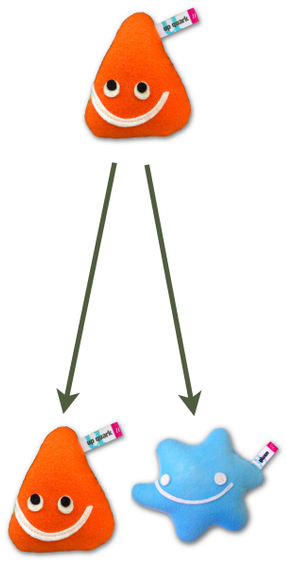
=



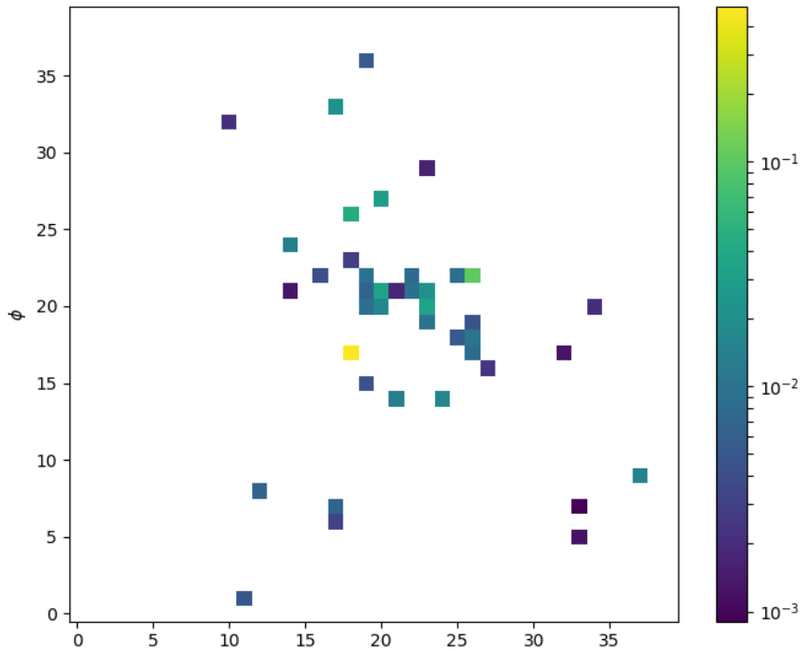


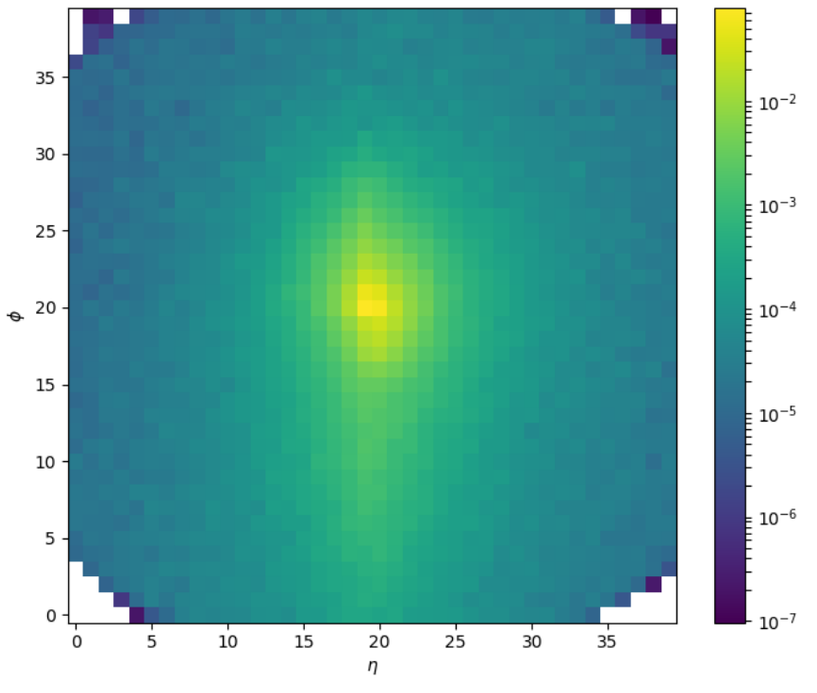
- *Binary classification task*
- *Fully supervised learning (using simulation)*
- *40x40 Pixels, E_T*
- *Perfectly suited for deep learning algorithms*

QCD Jet

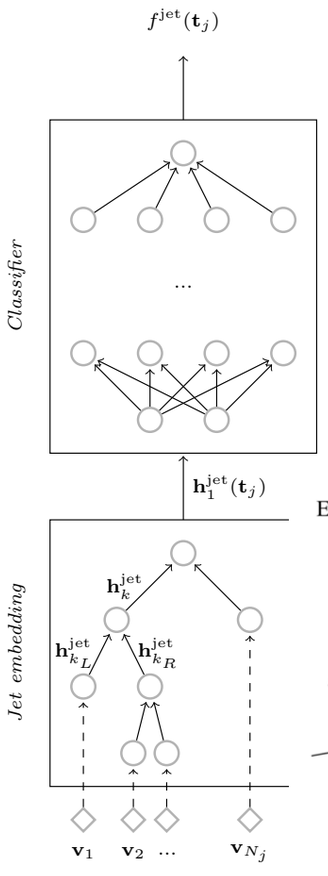


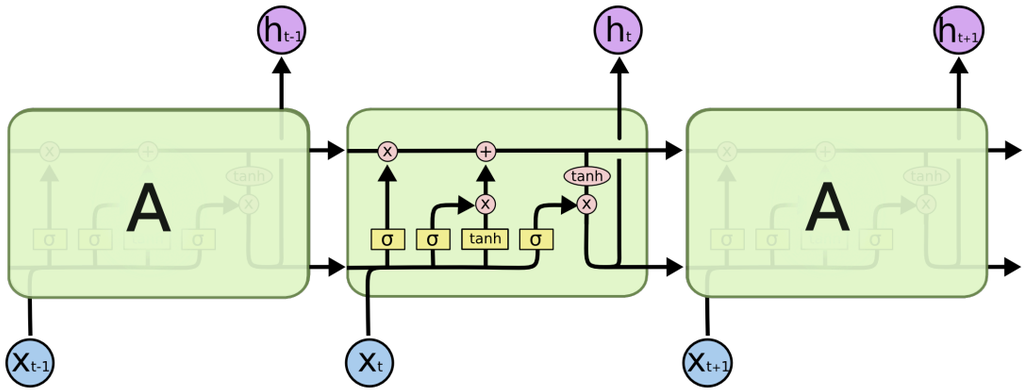
=

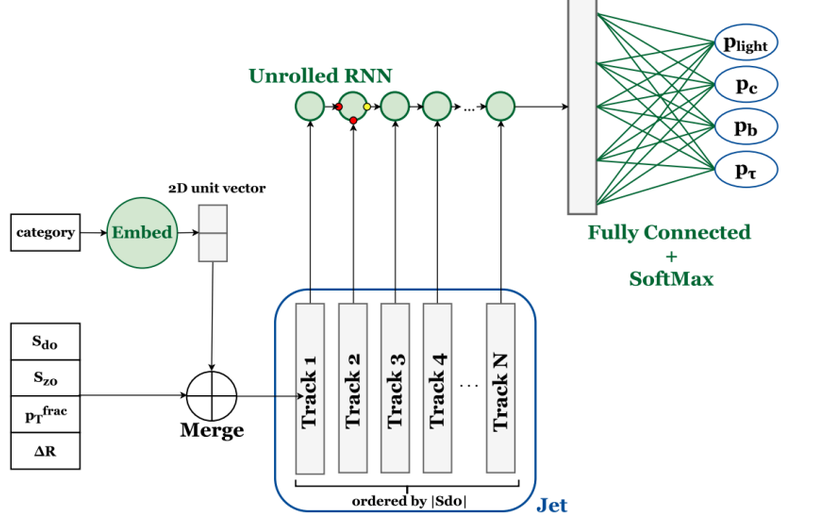


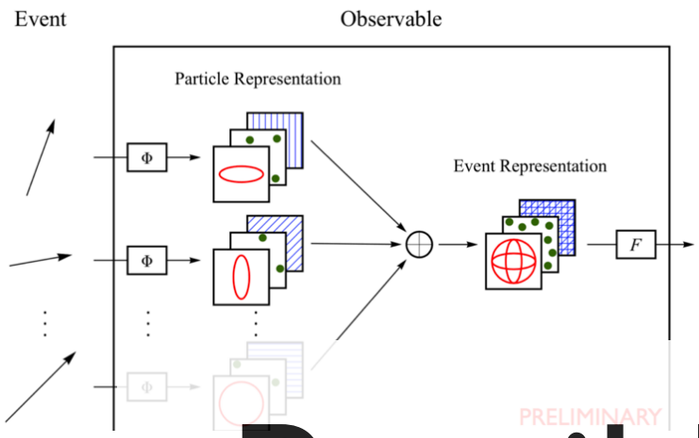


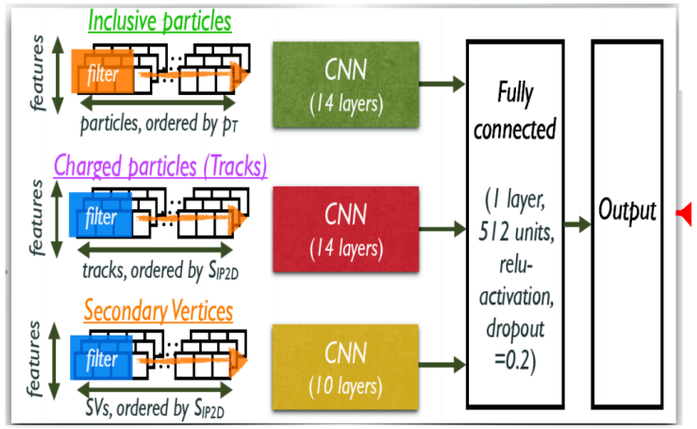
Possible Approaches

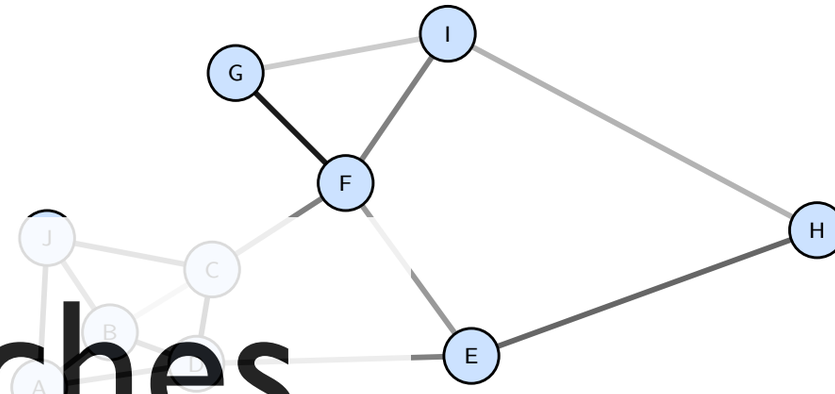


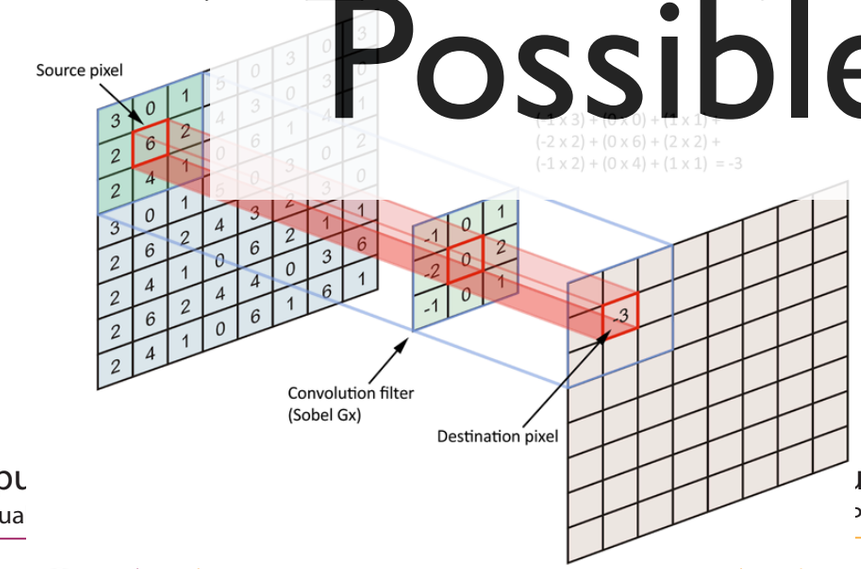


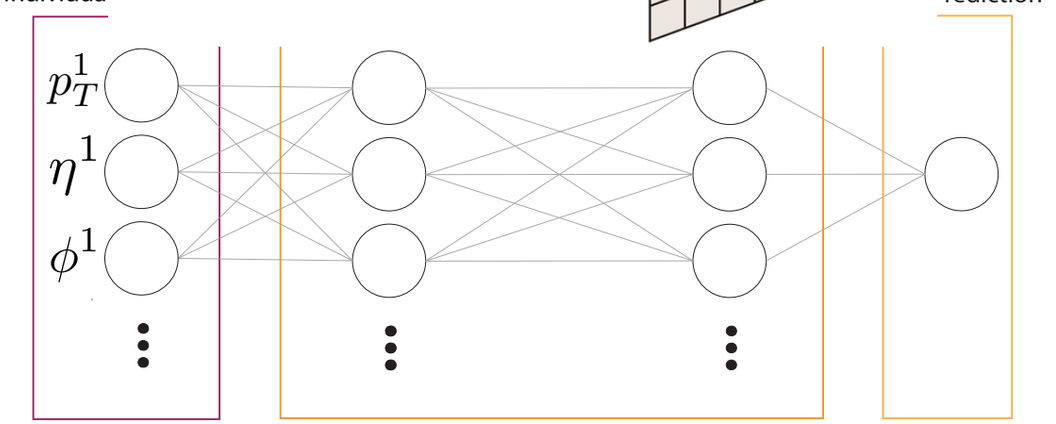












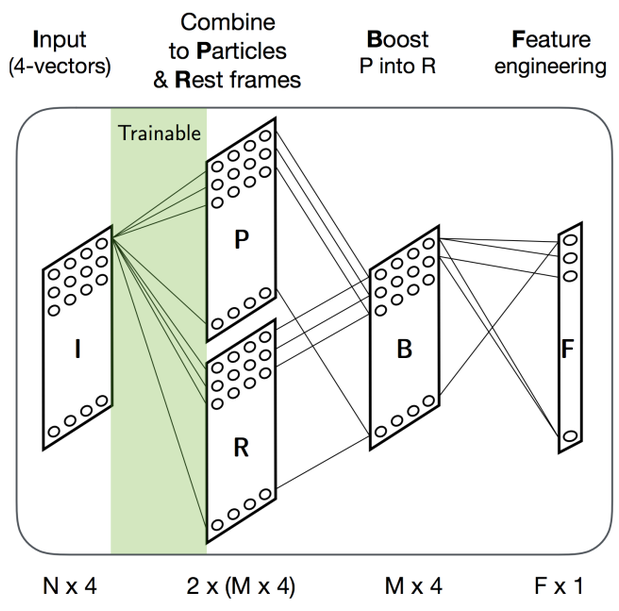
$$k_{\mu,i} = \begin{pmatrix} E_0 & E_1 & \dots & E_N \\ p_{x,0} & p_{x,1} & \dots & p_{x,N} \\ p_{y,0} & p_{y,1} & \dots & p_{y,N} \\ p_{z,0} & p_{z,1} & \dots & p_{z,N} \end{pmatrix}$$

Combination Layer (**CoLa**): create linear combinations:

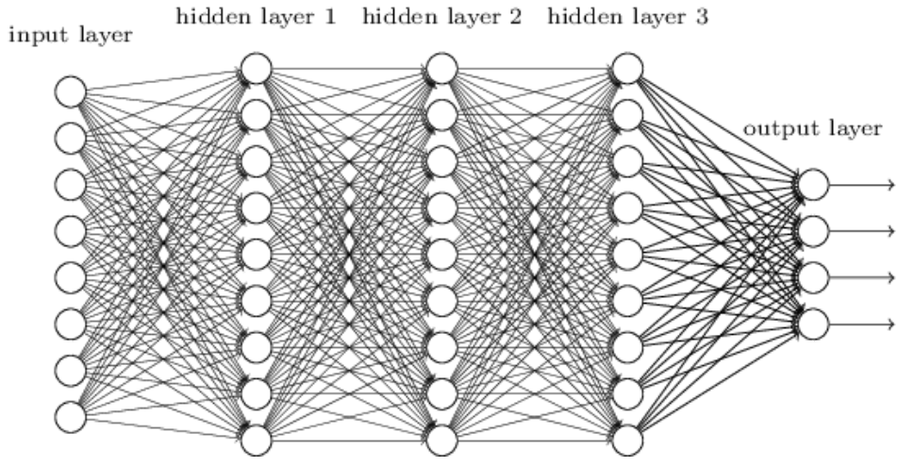
$$k_{\mu,i} \xrightarrow{\text{CoLa}} \widetilde{k}_{\mu,j} = k_{\mu,i}\, C_{ij}$$

Lorentz Layer (**LoLa**): Use resulting matrix to extract physics features.
Main assumption is the Minkowski metric

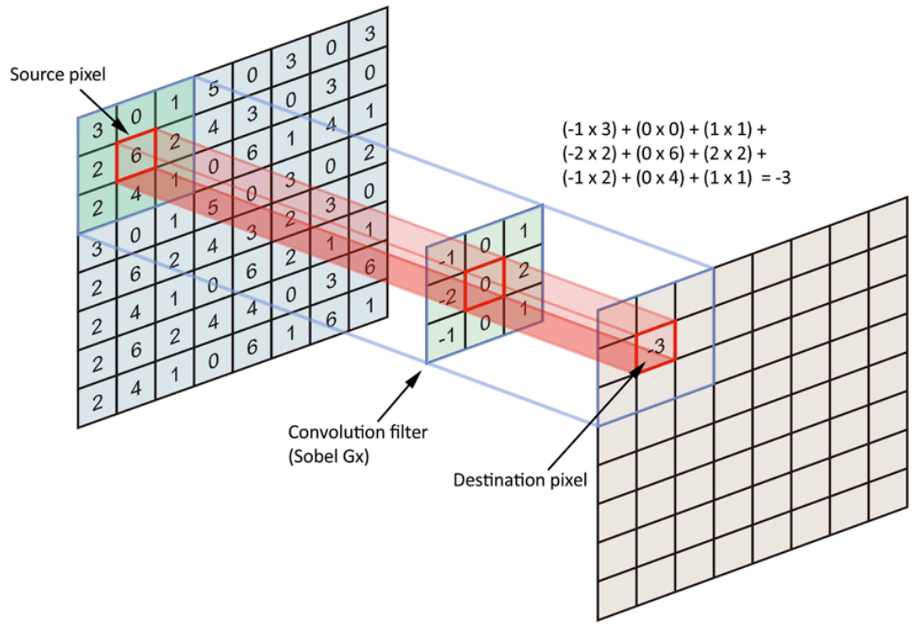
$$\widetilde{k}_{\mu,i} \to \sum_j (\widetilde{k}_i - \widetilde{k}_j)_\mu (\widetilde{k}_i - \widetilde{k}_j)_\nu \eta^{\mu\nu} B_{ij}$$



High-level: Fully Connected

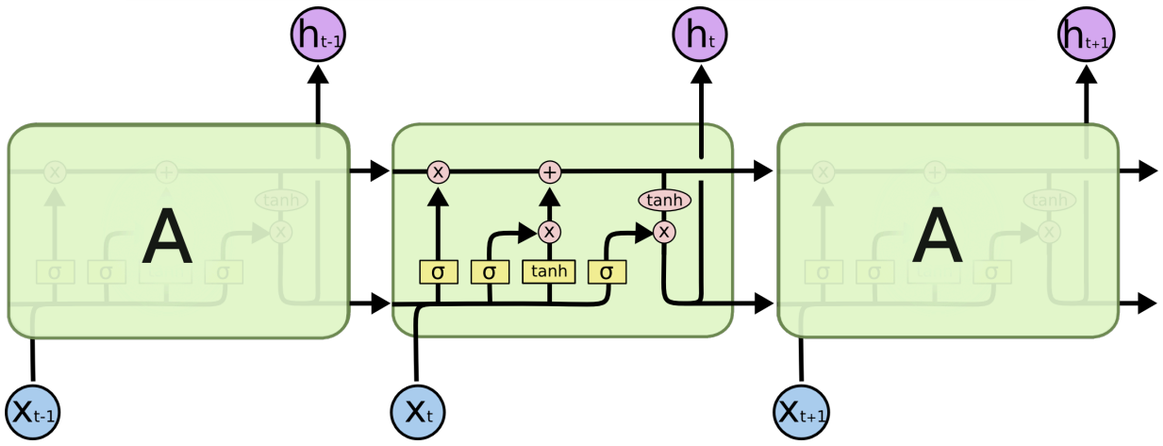


Regular grid: Convolution

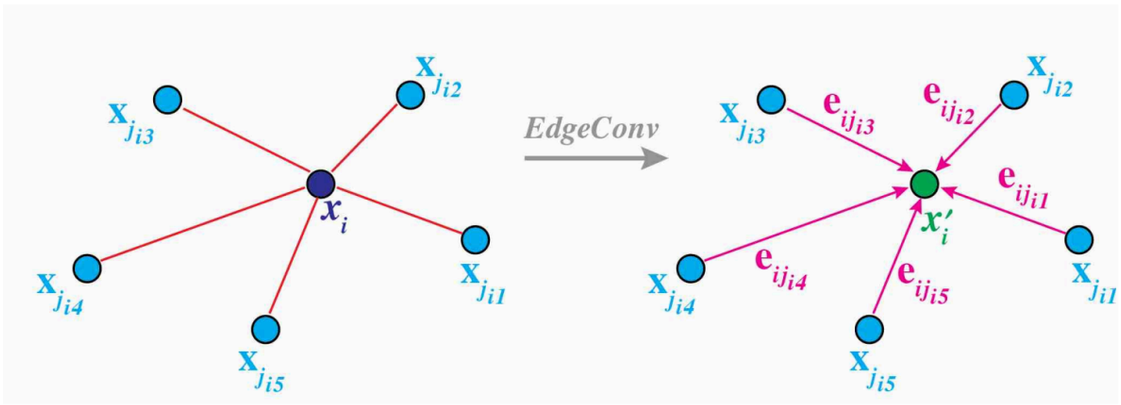


Representation

Time series: Recurrent

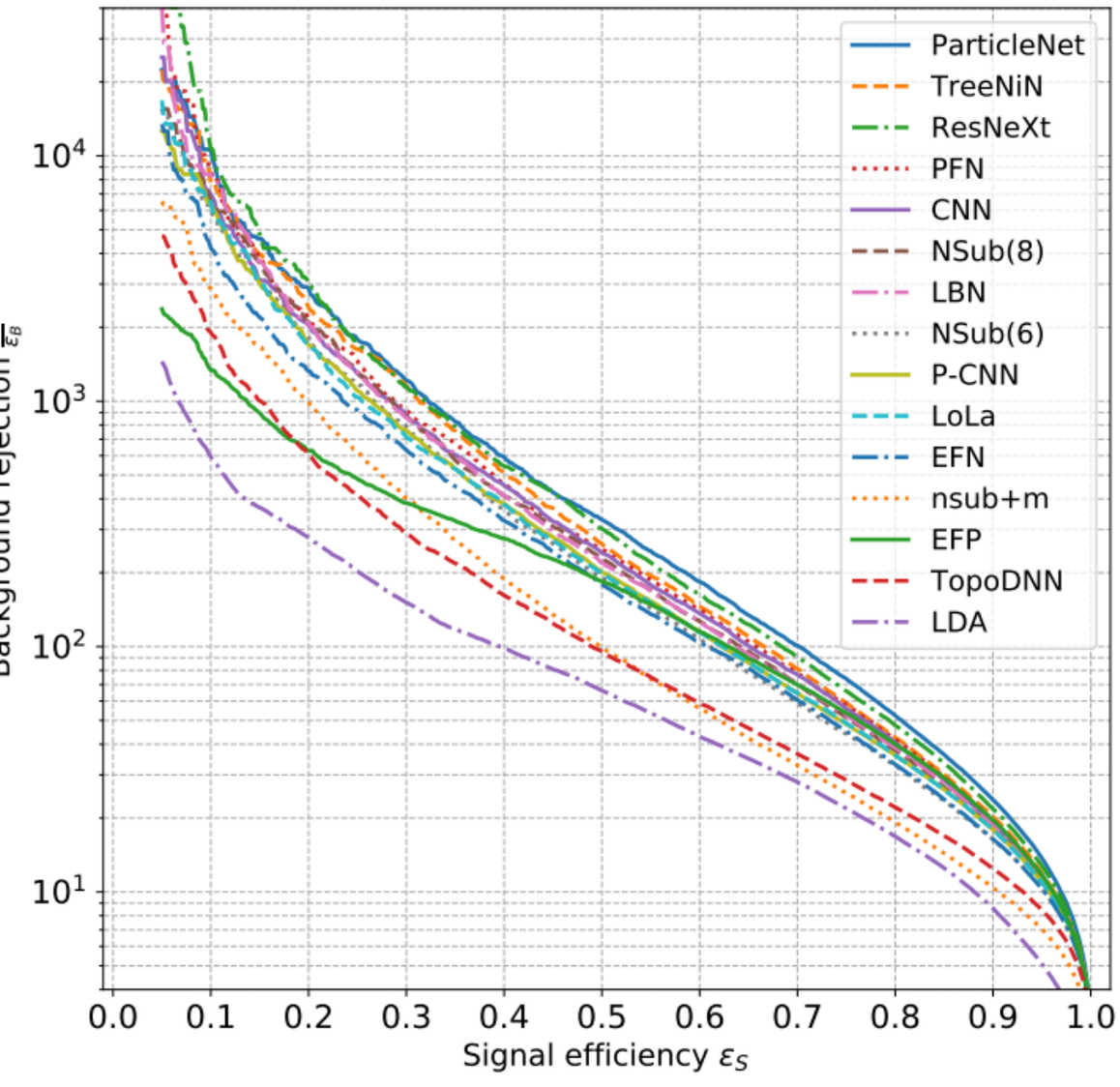


Point cloud: Sets & Graphs



Results

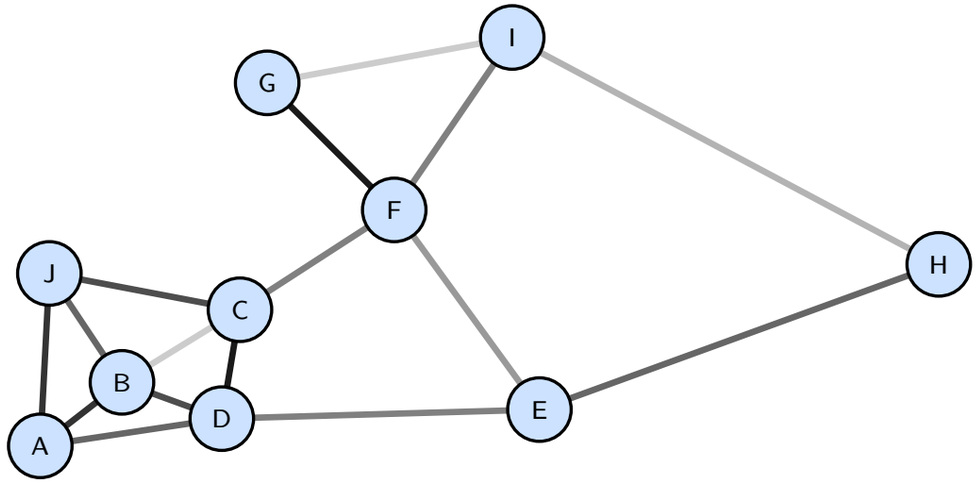
Community performance comparison (toy dataset public): 1902.09914



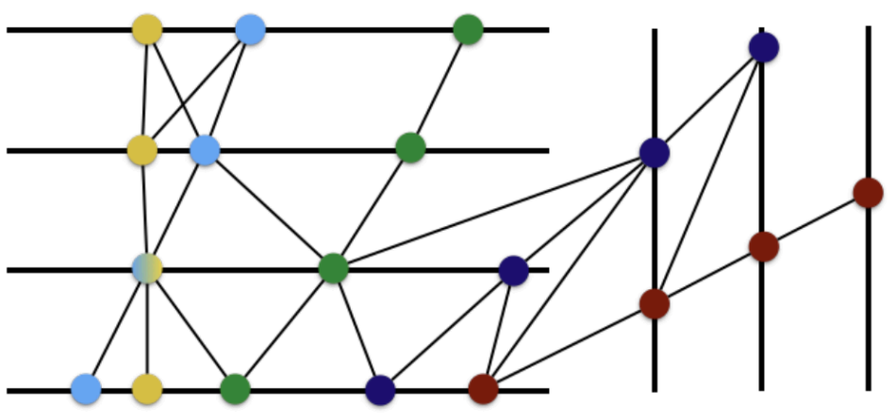
- Great test-bed to compare different data representations
  - (and, of course, useful for new physics searches, top/Higgs measurements)
- Still surprising gains in performance
  - Although it needs to be seen how well these translate to data
- (Also developments in flavour tagging, not covered here)

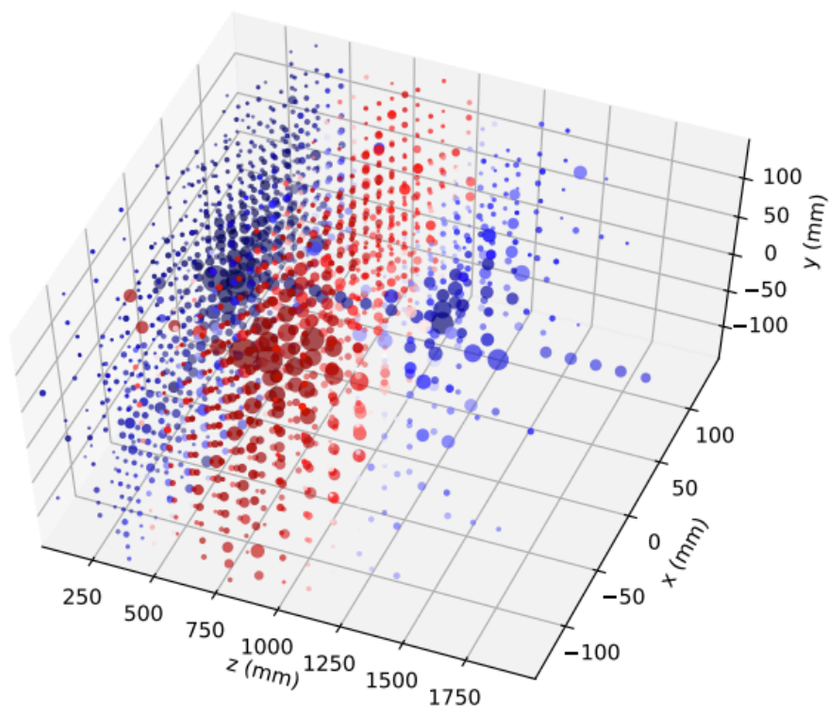
ParticleNet = Graphs

- Images are a convenient representation, but do not capture real structure of our measurements
- Alternative: Graphs
  - Vertex: Particle
  - Edge: Distance (for example geometric)



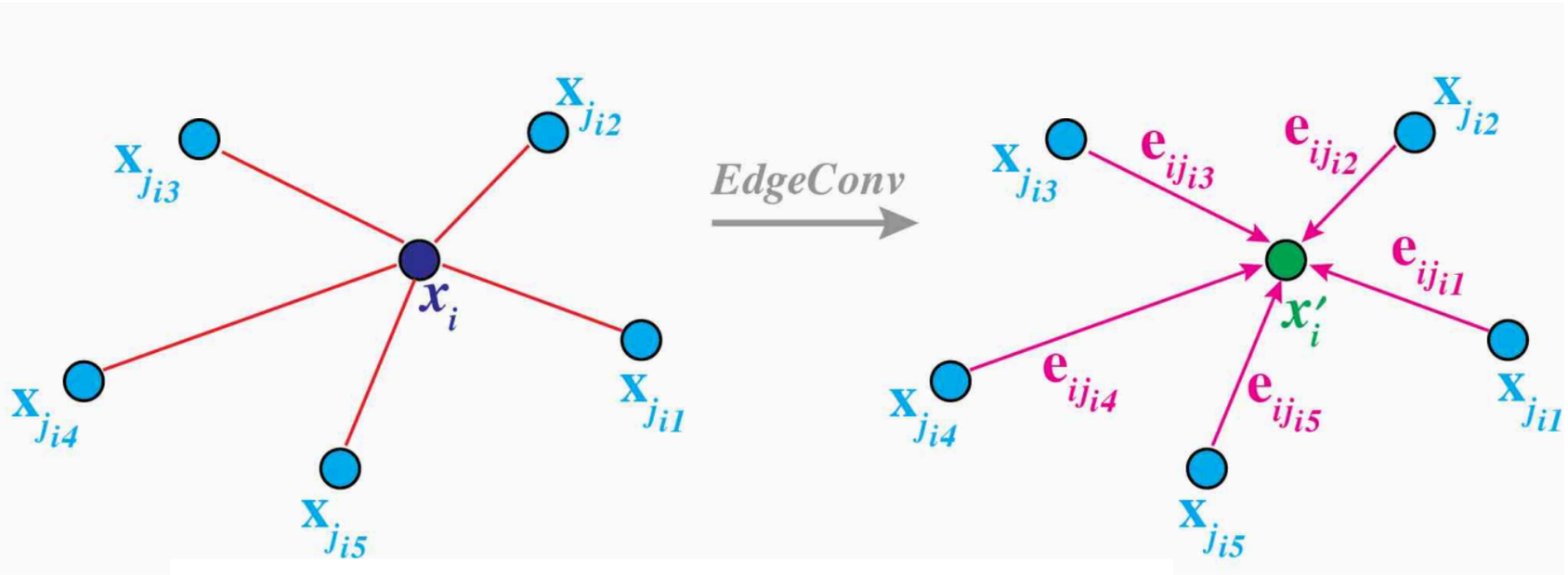
- Active development of graphs on CS side, but already HEP applications:
  - Particle Net (best performing top tagger in community s based on EdgeConv) (1902.08570)
  - Calorimeter Clustering (1902.07987)
  - Tracking (1810.06111)

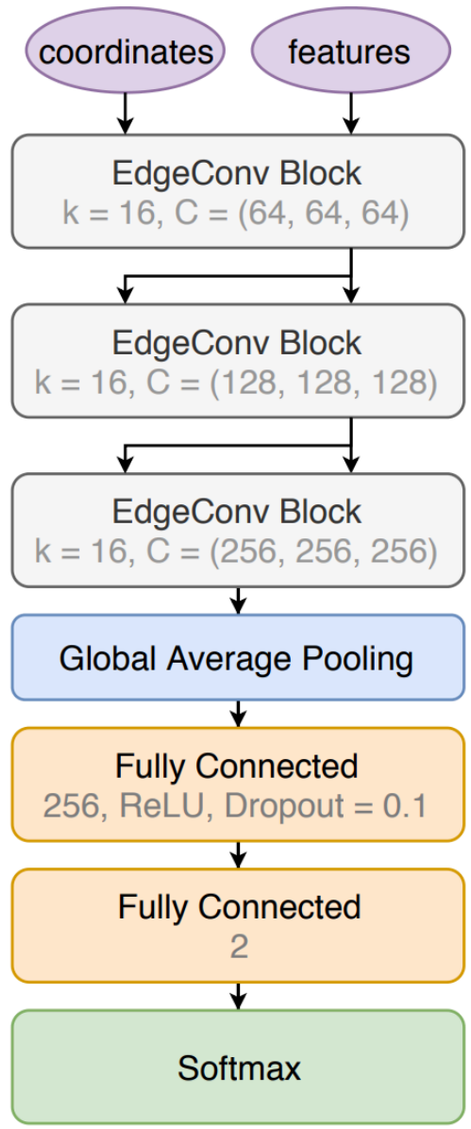




1902.08570
1801.07829

Closer look



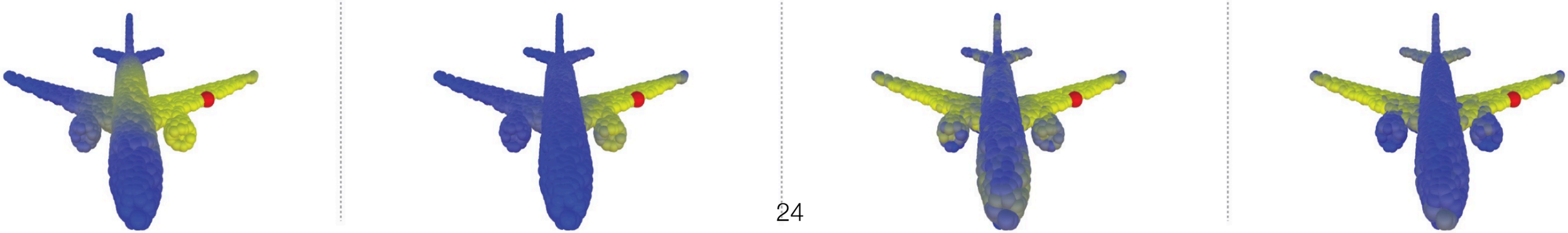


$$x'_i = \square_{j=1}^{k} h_\Theta(x_i, x_{i_j})$$

Aggregation function (sum or max)

Neural network

$$h_\Theta(x_i, x_{i_j}) = \bar{h}_\Theta(x_i, x_{i_j} - x_i)$$



Generative models

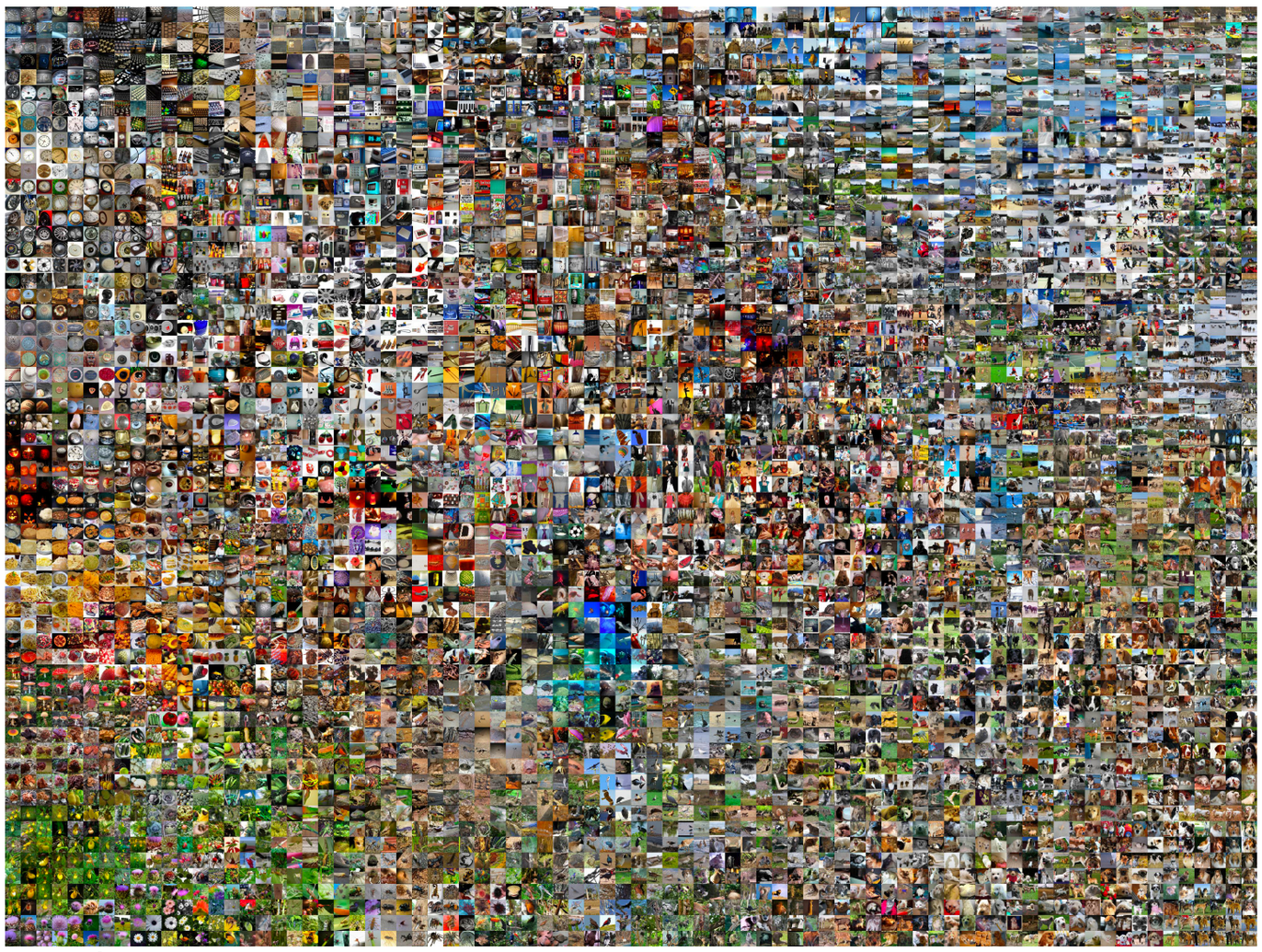
Generative Networks

Large Scale GAN Training for High Fidelity Natural Image Synthesis, A Brock, J Donahue, K Simonyan, 1809.11096

https://www.thispersondoesnotexist.com/

Generators

We **have**:
many images
(or collision events,
or detector readouts, …)



wired.com

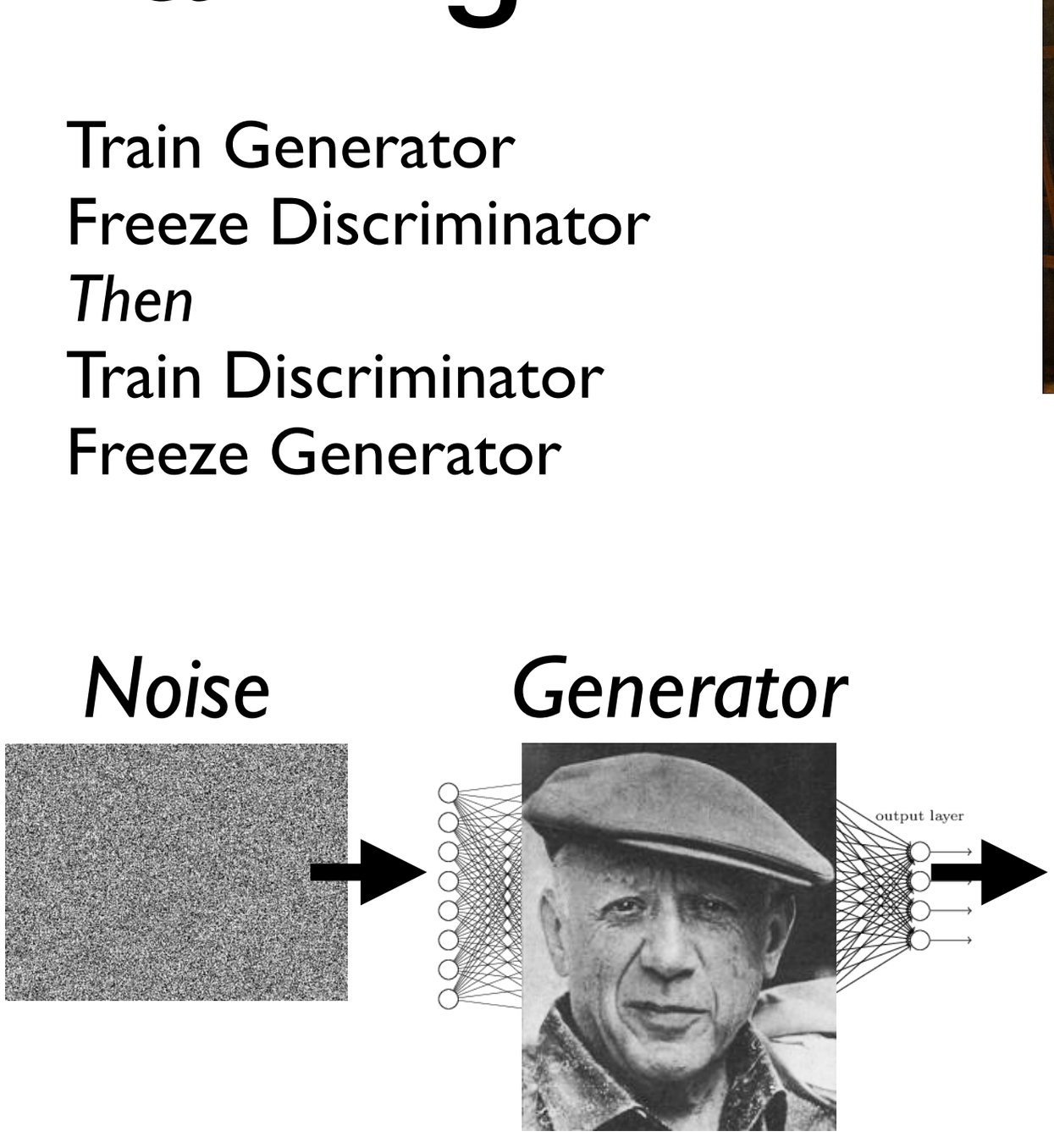
We **want**: more images.

(Specifically: New examples that are similar to the examples, but not exact copies)

How to encode in
neural net?

Alternating Training

Train Generator
Freeze Discriminator
Then
Train Discriminator
Freeze Generator

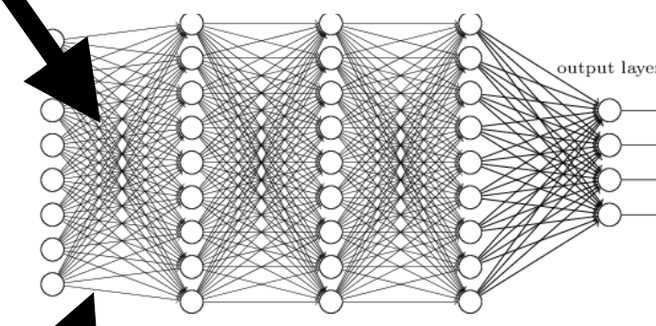


Alternating Training

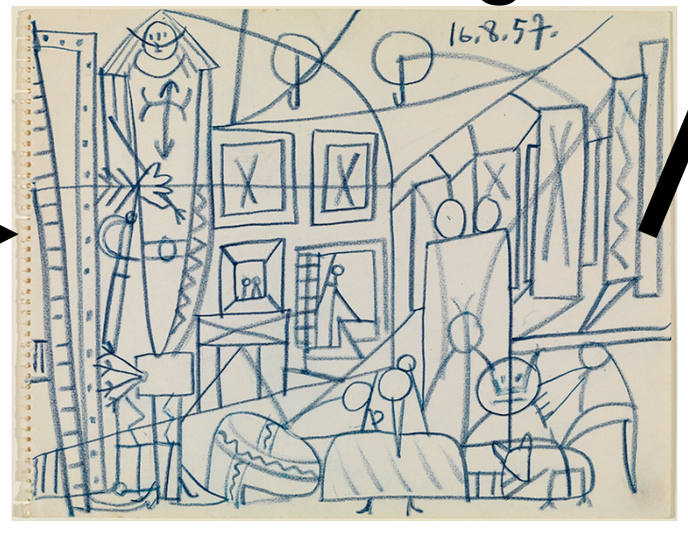
Train Generator
Freeze Discriminator
Then
Train Discriminator
Freeze Generator

Real Images

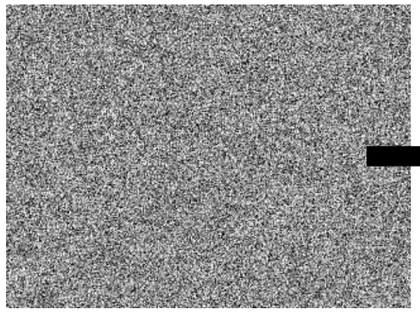
Discriminator



Fake Image



Noise



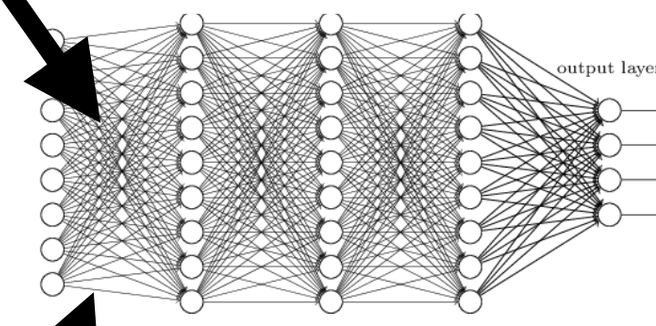
Generator

Alternating Training

Train Generator
Freeze Discriminator
Then
Train Discriminator
Freeze Generator

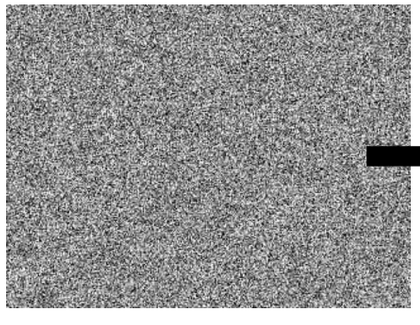
Real Images

Discriminator

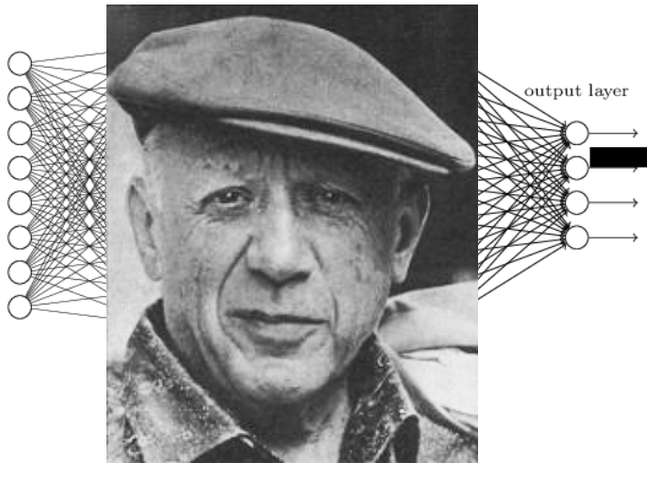


Fake Image

Noise



Generator

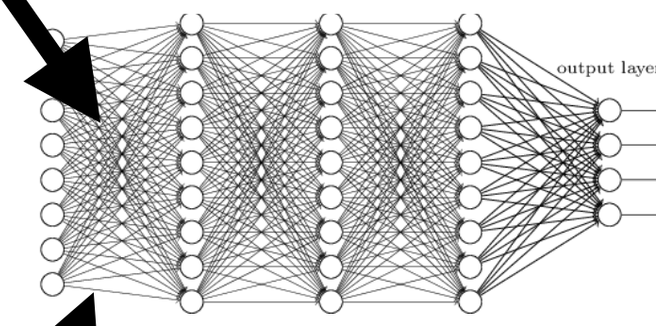


Alternating Training

Train Generator
Freeze Discriminator
Then
Train Discriminator
Freeze Generator

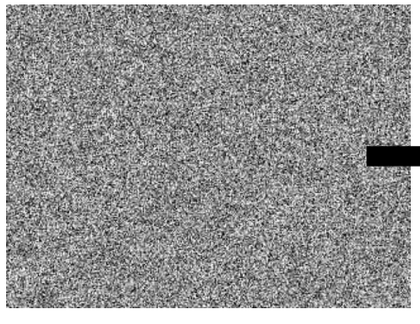
Real Images

Discriminator



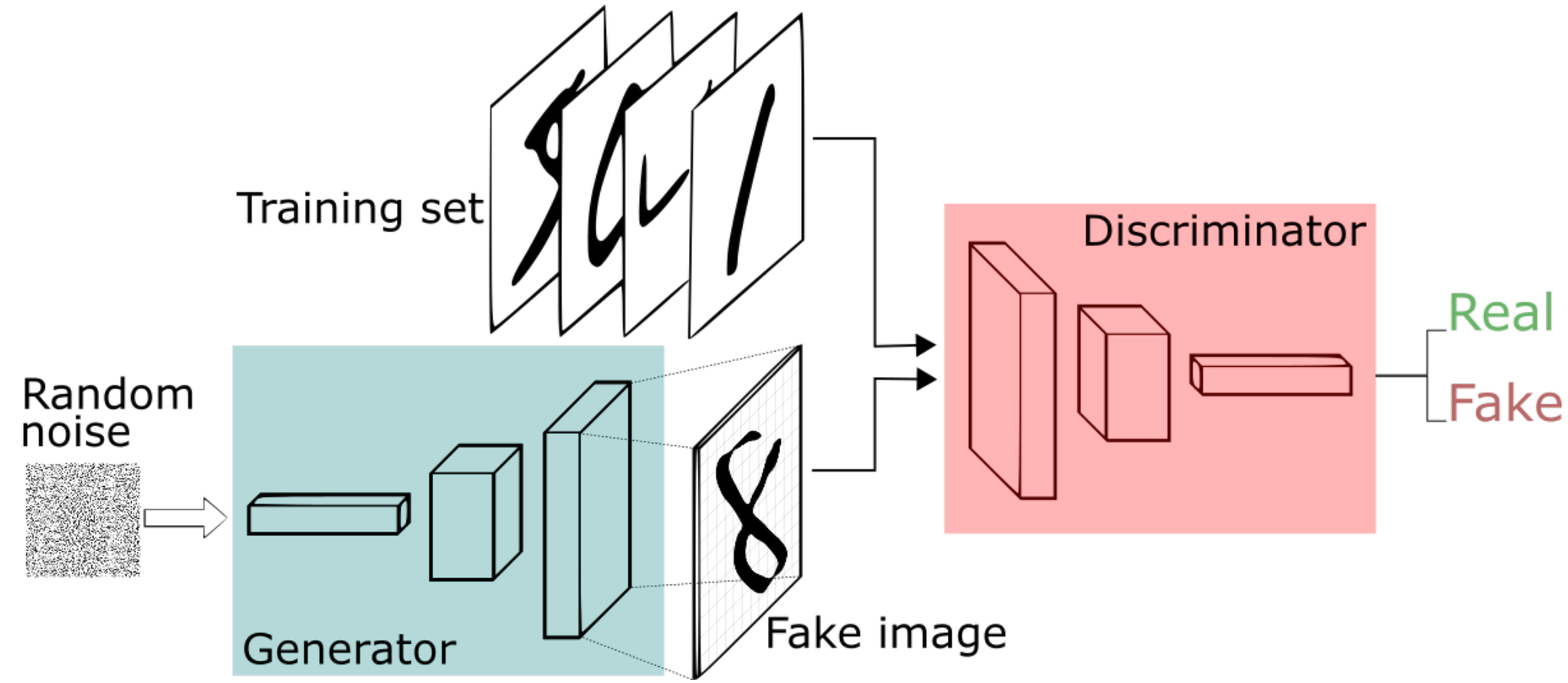
Fake Image

Noise



Generator

Generative Adverserial



$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

Real Samples

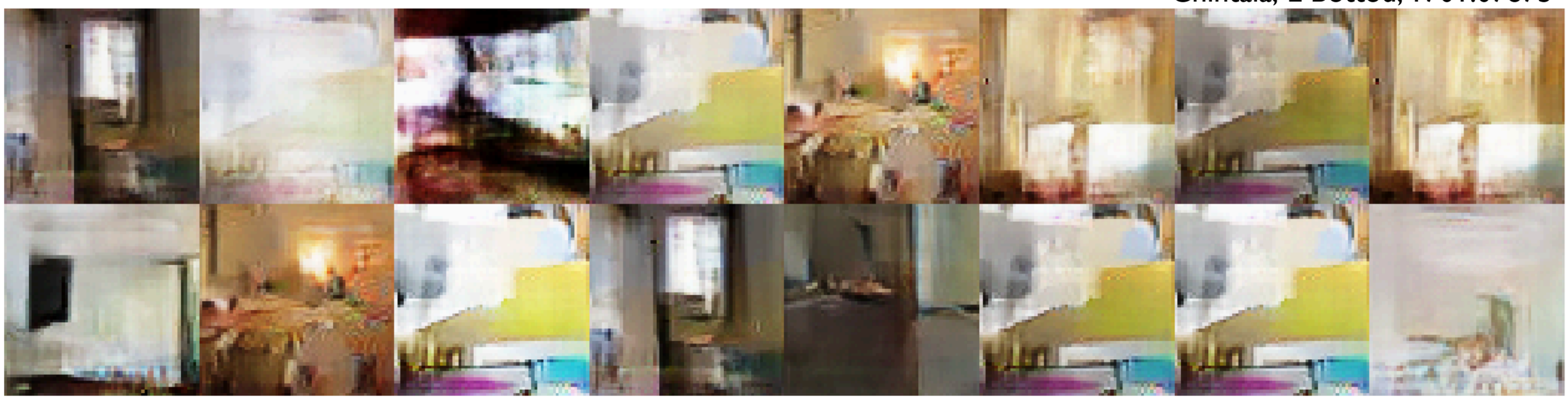
Generated Fake Samples

Generative Adversarial Networks
IJ Goodfellow et al
1406.2661

GAN Problems

- Stability and convergence of learning
- Generator & Discriminator matching
  - Vanishing gradient
  - (use small momentum in training)
- Mode collapse
- Hard to interpret loss
  - Not correlated to image quality
- Similar to issues with adversarial training

lilianweng
Wasserstein GAN, M Arjovsky, S Chintala, L Bottou, 1701.07875

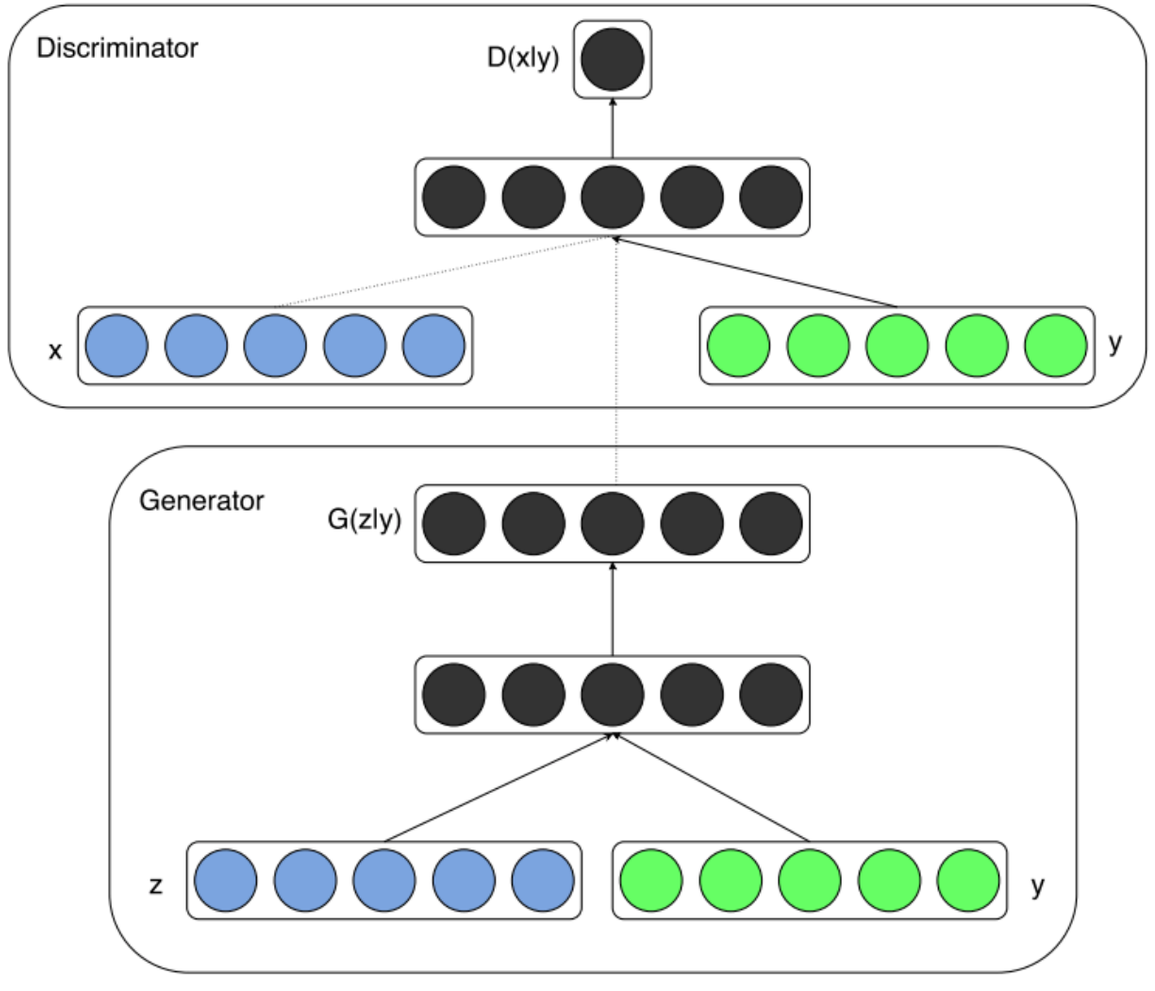


Label Conditioning

- To improve usefulness (and training) of GANs:
  - Provide information on picture we are simulating (label y)
  - Use this information in training of generator and discriminator conditioning

$$\log D(x) + \log(1 - D(G(z)) \rightarrow \log D(x|y) + \log(1 - D(G(z|y'))$$

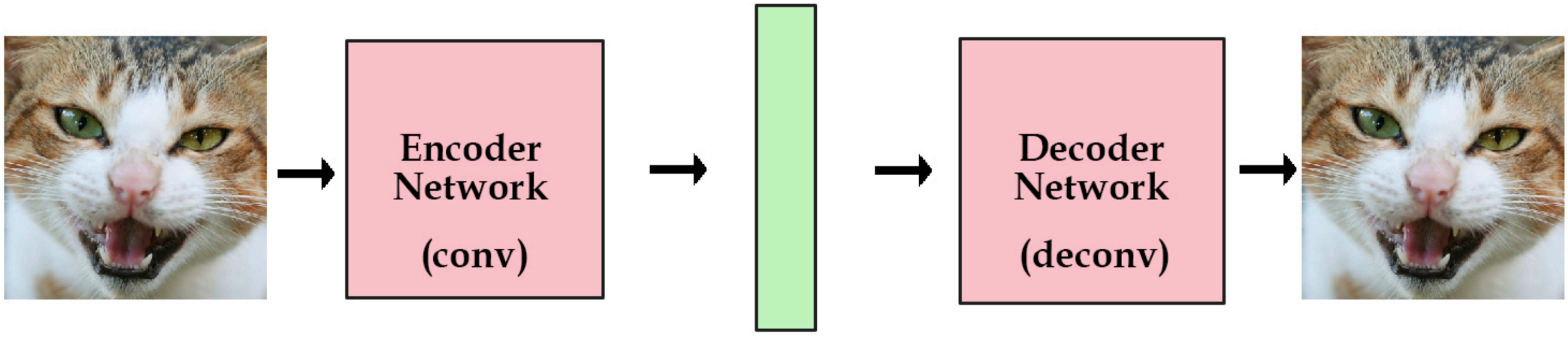
- Counteract mode collapse
- Key for physics application
  (labels: energy, particle type, …)



Conditional Generative Adversarial Nets
M Mirza, S Osindero
1411.1784

Variational Autoencoder

Autoencoder

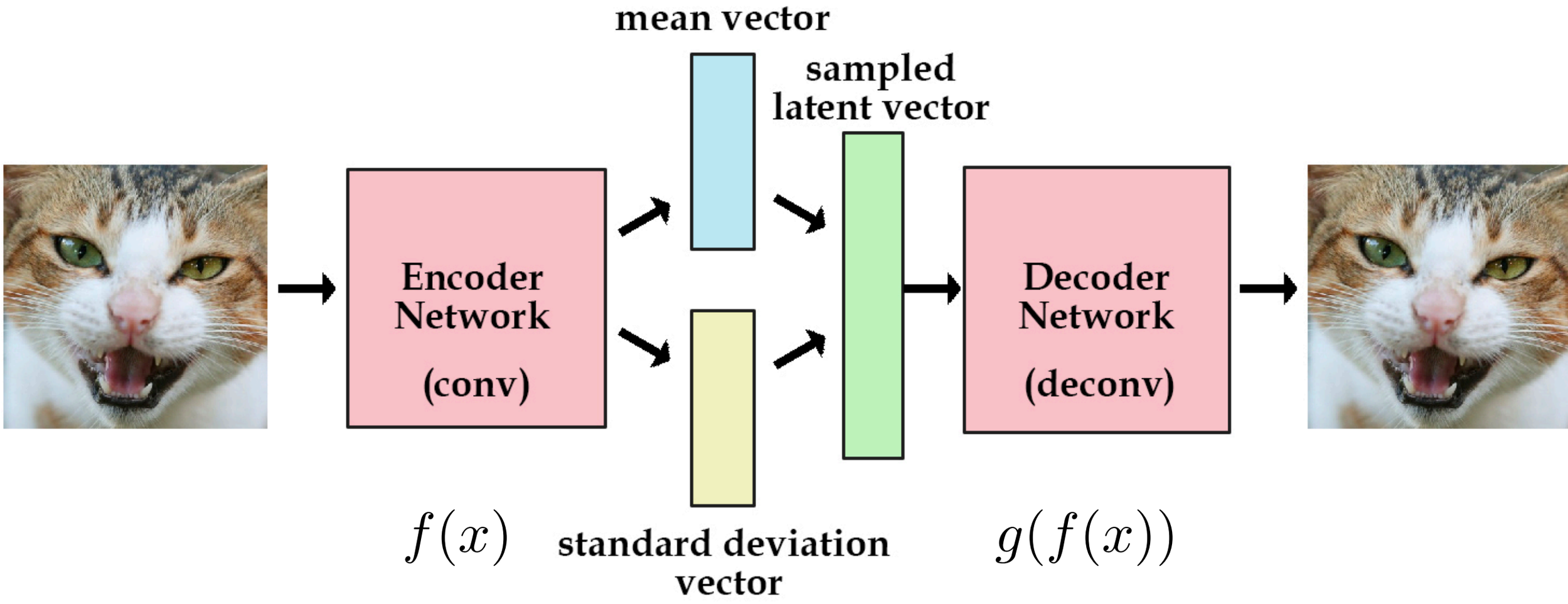


$f(x)$ **latent vector / variables** $g(f(x))$

$$L = (\hat{y} - g(f(x)))^2$$

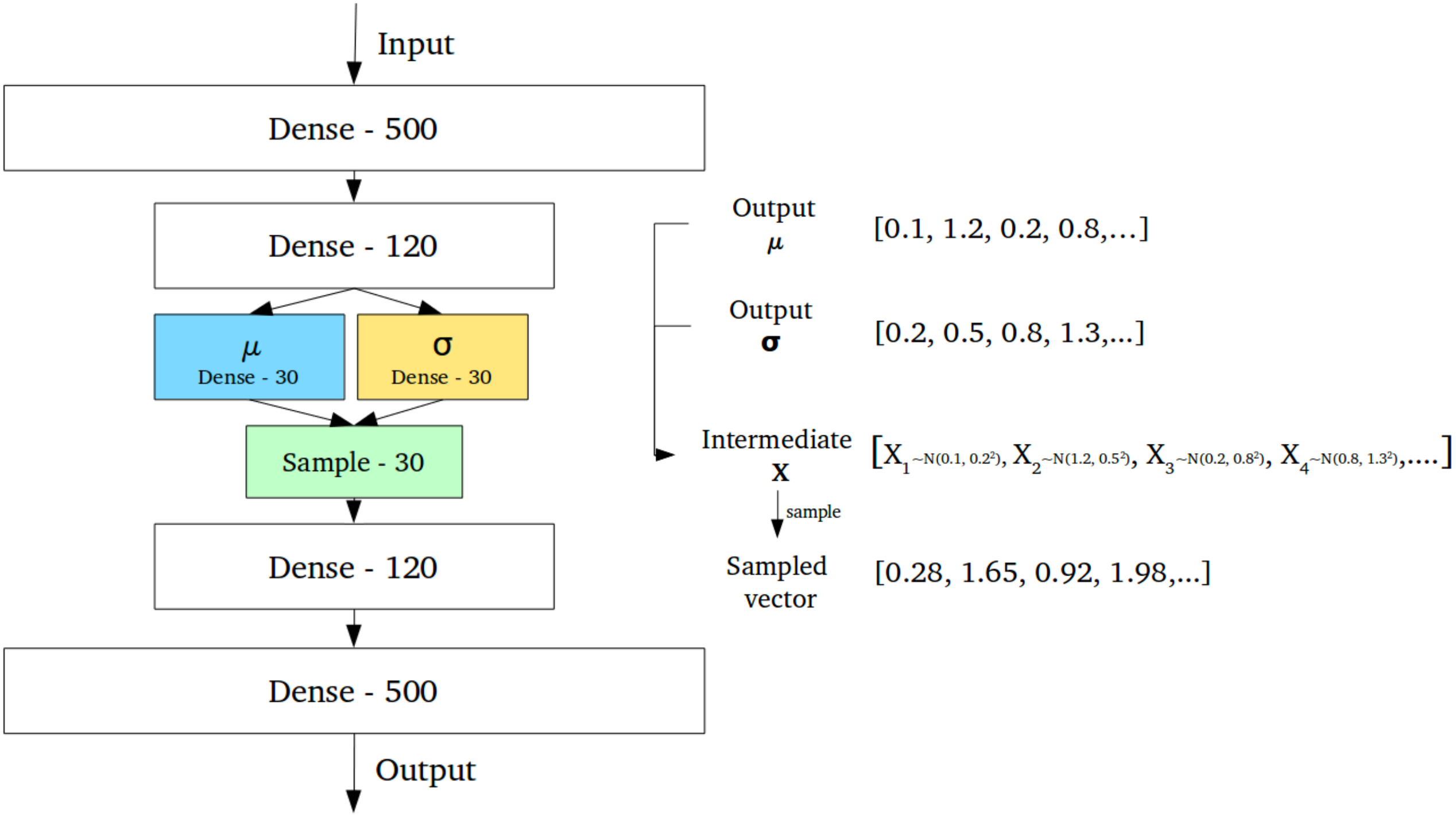
- Self-supervised learning
- *Latent space/bottleneck* with compressed representation (remember yesterday!)
- Dimension reduction
- Denoising
- Anomaly detection (later today!)

Variational Autoencoder

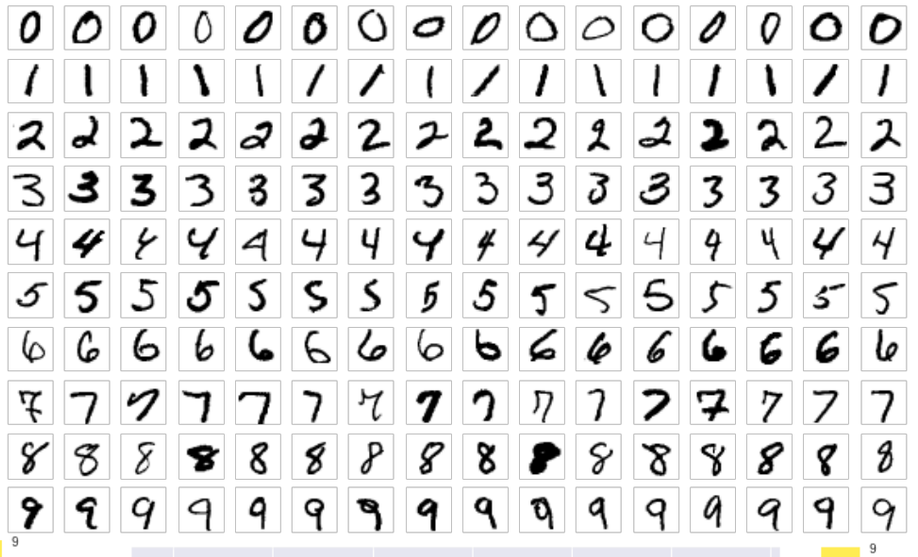


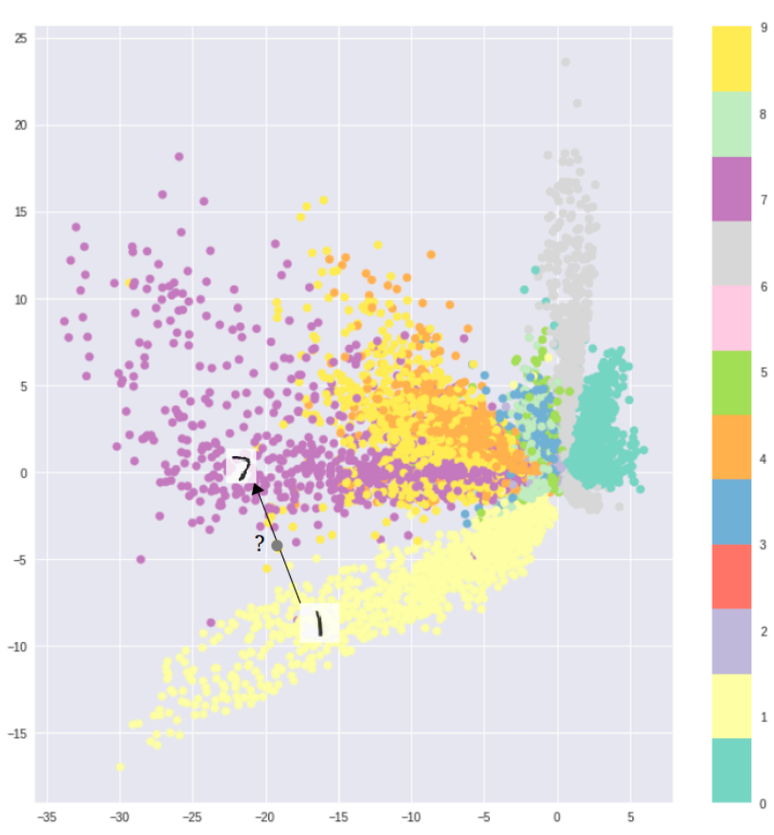
- Want to sample from latent space
- Split into mean and standard deviation
- Add penalty term (Kullback-Leibler divergence) so mean/std are close to unit Gaussian

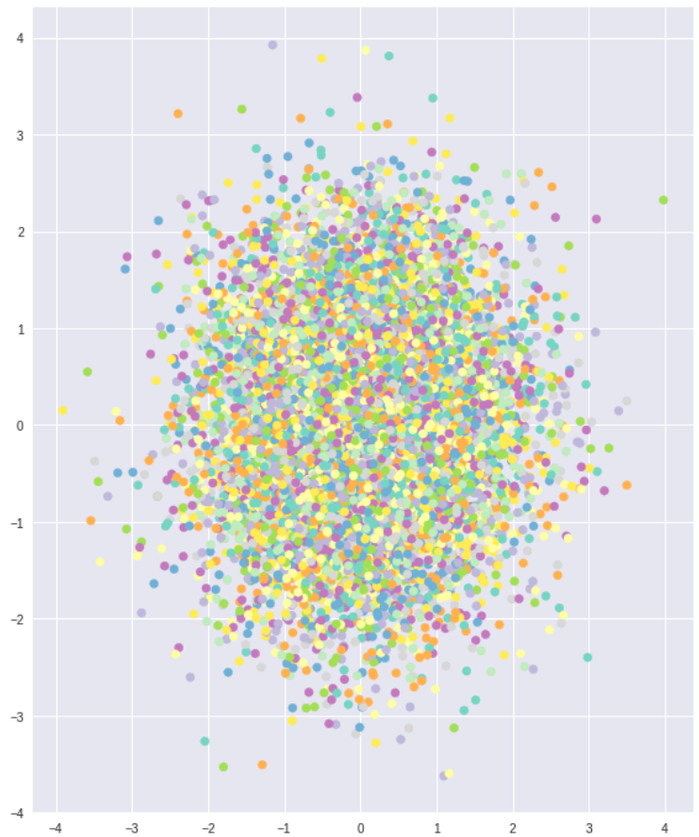
Concrete

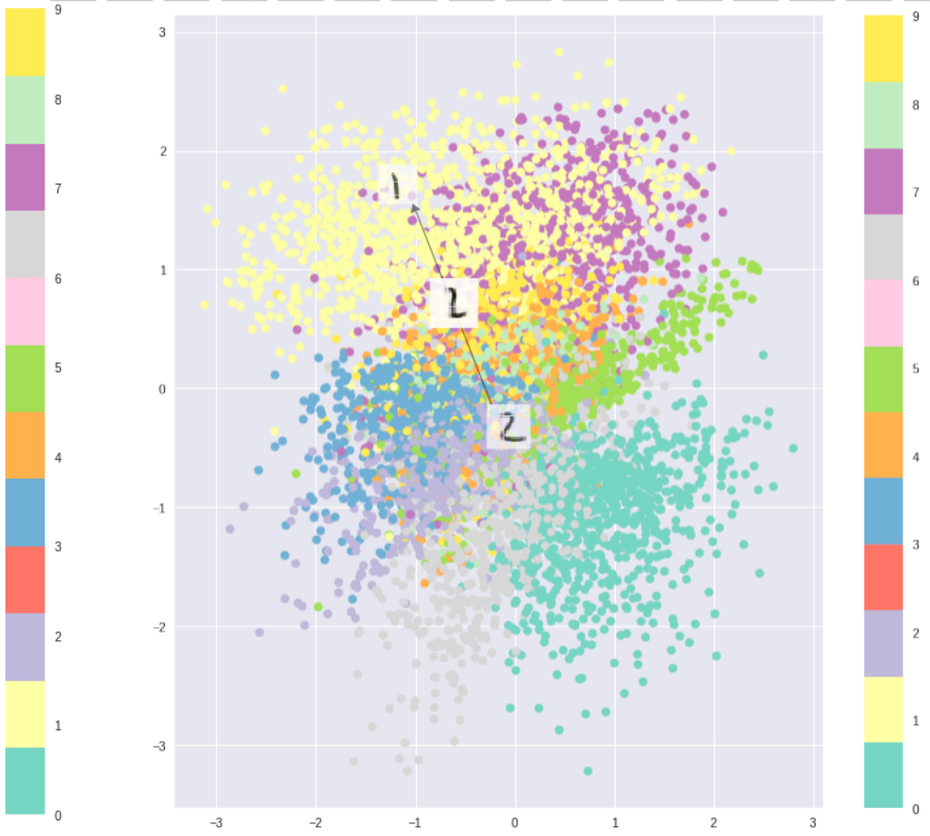

Input
Dense - 500
Dense - 120
μ
Dense - 30
σ
Dense - 30
Sample - 30
Dense - 120
Dense - 500
Output
Output
μ
[0.1, 1.2, 0.2, 0.8,...]
Output
σ
[0.2, 0.5, 0.8, 1.3,...]
Intermediate
X
[X1~N(0.1, 0.2²), X2~N(1.2, 0.5²), X3~N(0.2, 0.8²), X4~N(0.8, 1.3²),....]
sample
Sampled
vector
[0.28, 1.65, 0.92, 1.98,...]

Variational Autoencoder









Reconstruction Loss Only

$$L = (\hat{y} - g(f(x)))^2$$

KVfrans

1312.6114

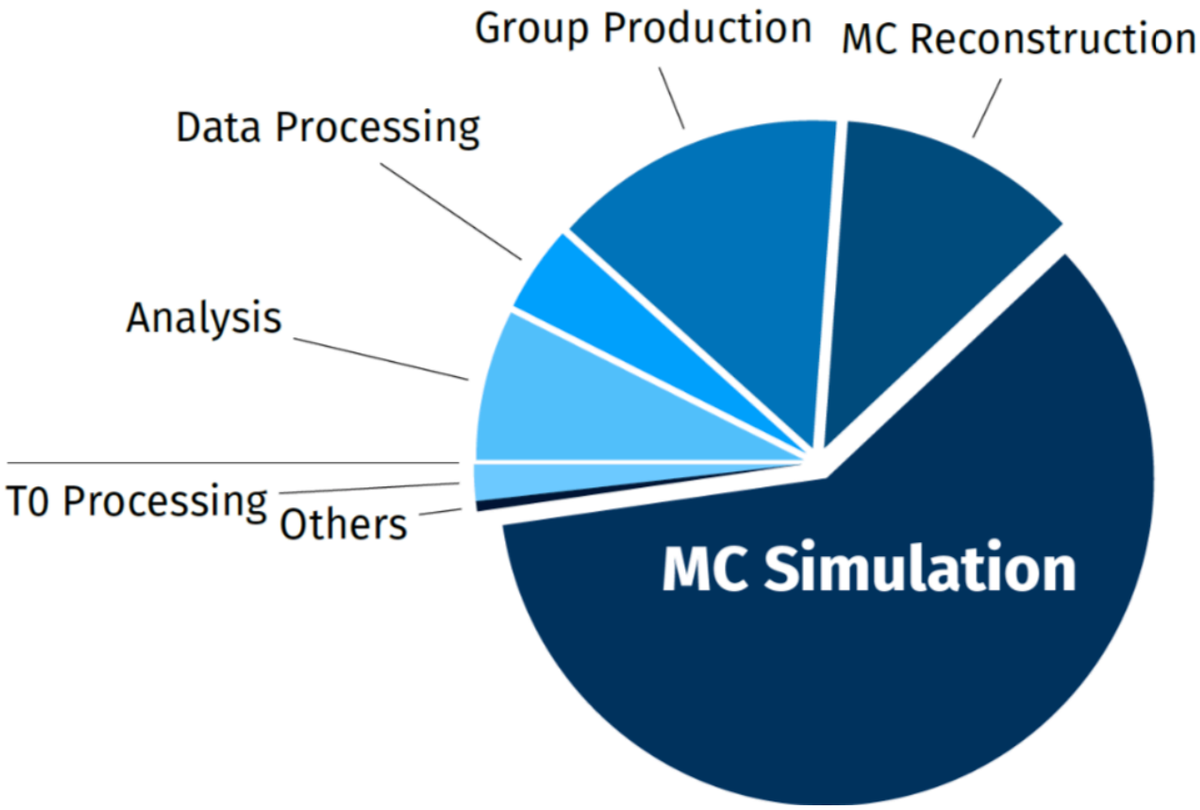
KL Loss Only

$$\frac{1}{2}\sum_{j=1}^{J}\left(1+\log((\sigma_j)^2)-(\mu_j)^2-(\sigma_j)^2\right)$$

Combined Loss

towardsdatascience.com

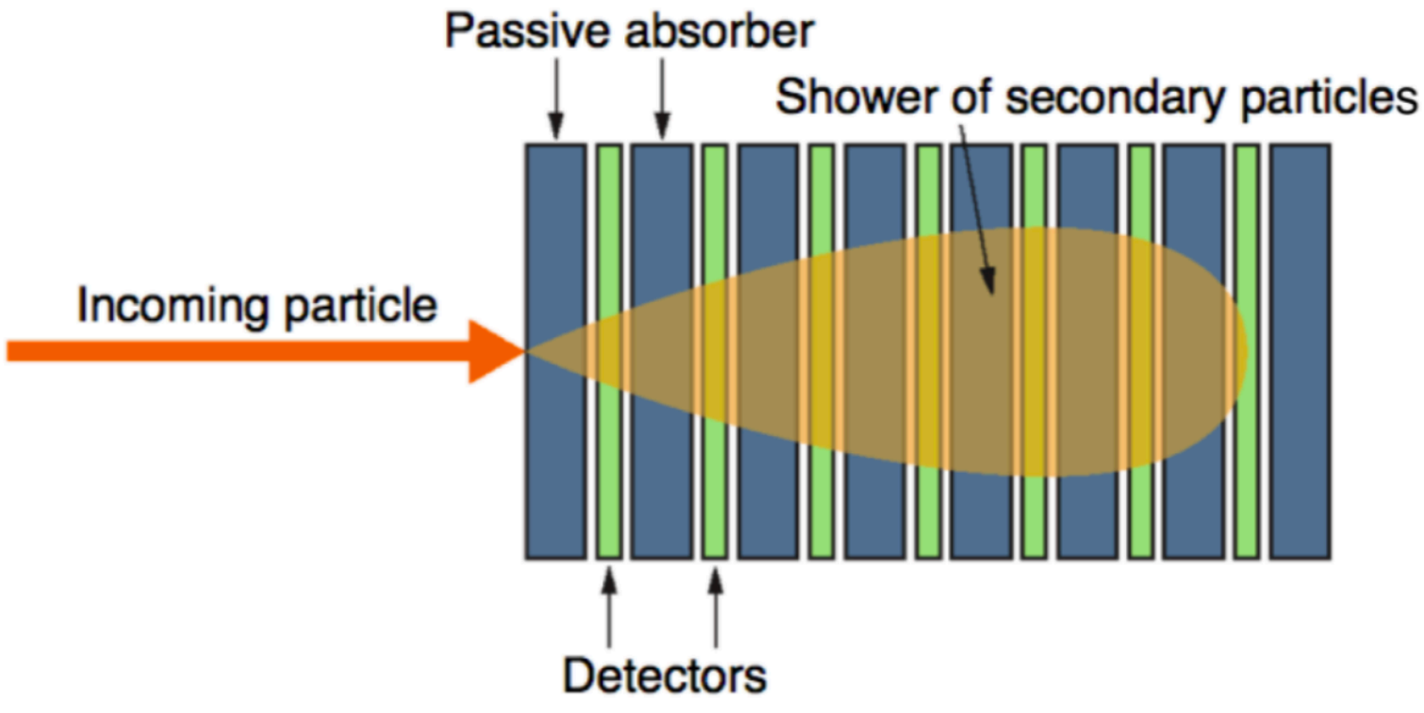
Physics Uses

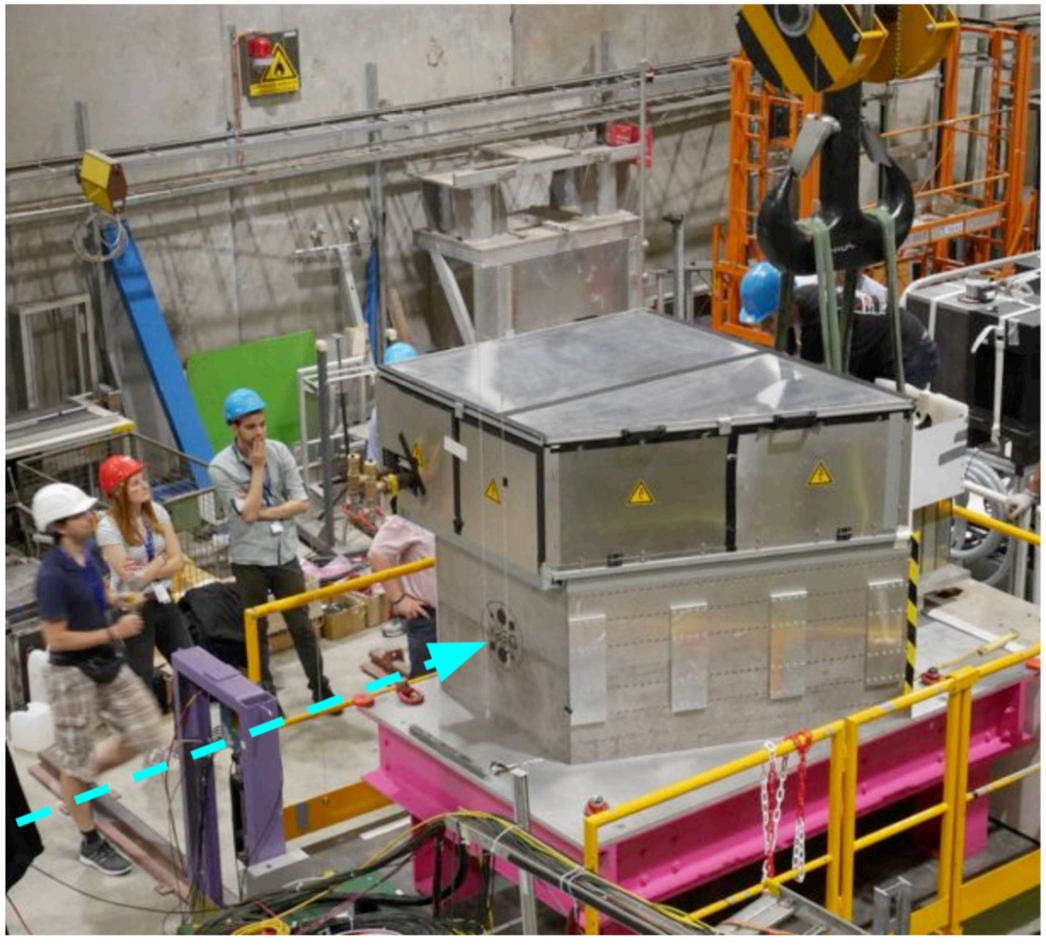


Tobias Golling, Hammer&Nails 19

Particle Showers

Main motivation:
Fast simulation of interaction between particles and detector material
Started by CaloGAN (1705.02355)

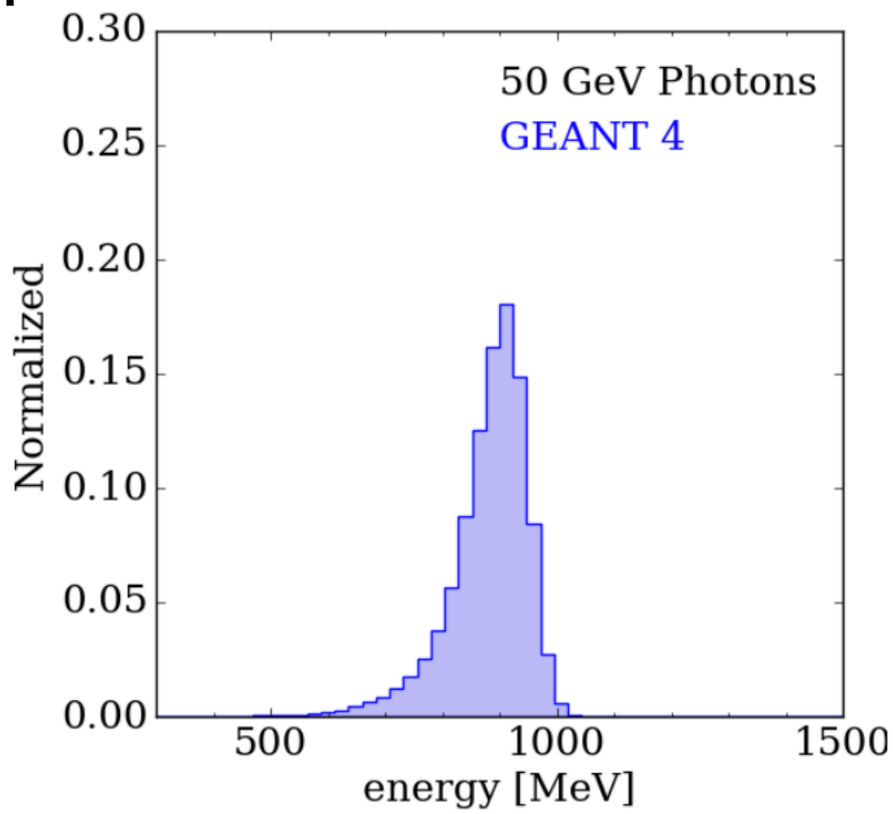




Generative models are also applied to:
phase space integration and sampling, event generation, ….

Additional Challenges

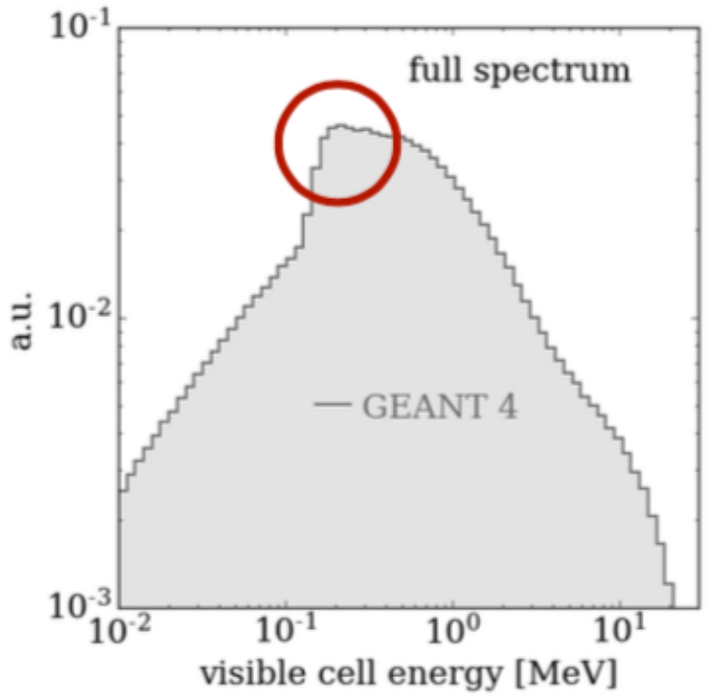
- How to evaluate convergence of models?
- Correctly model differential distributions
- Condition on a large number of quantities (energy, particle type, impact position, angle, …)
- Other considerations:
  Coverage (do I produce example for all phase space?)
  Saliency (is this a good example of the desired type of event)
  Mode collapse
  Overfitting

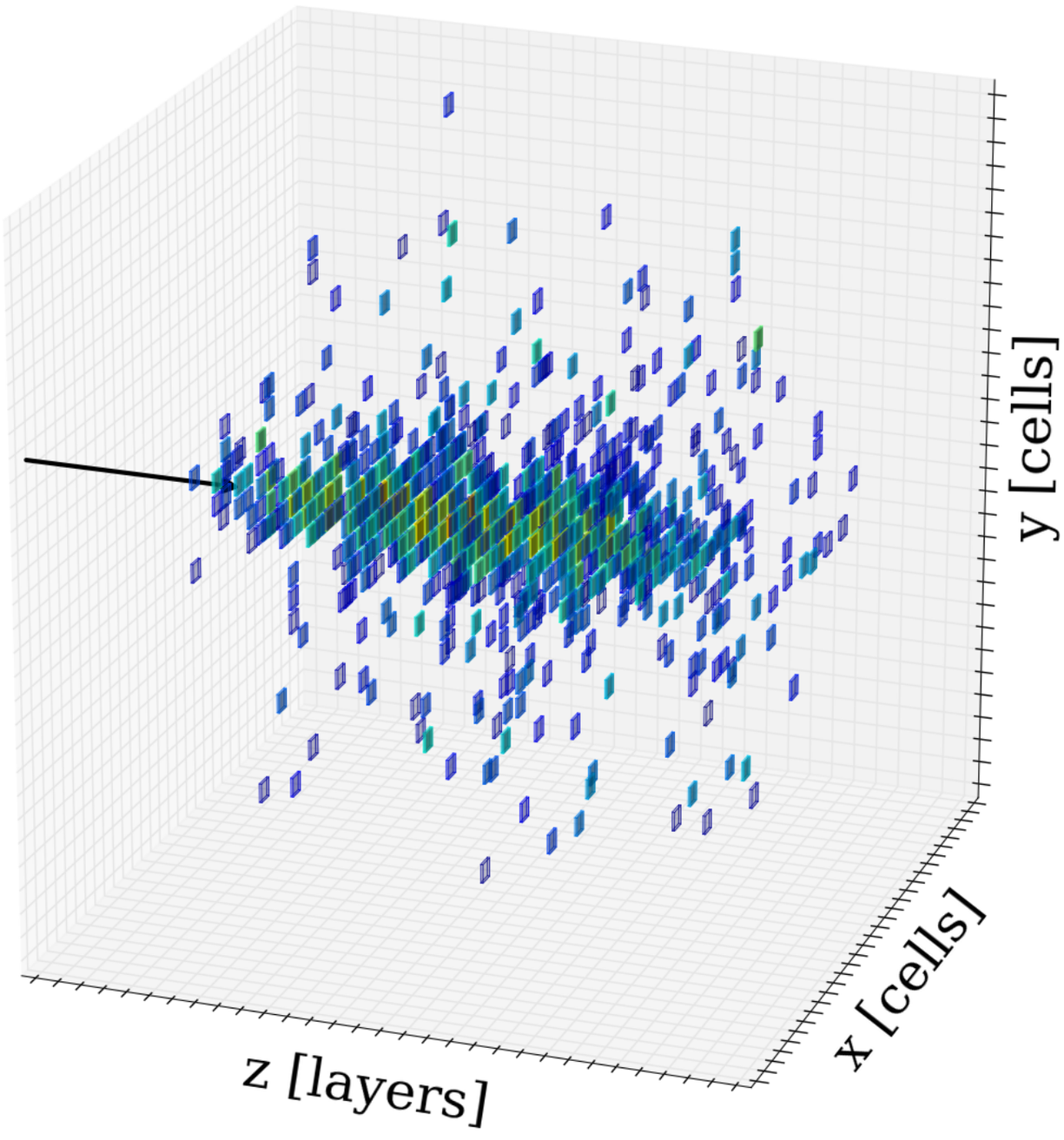


Concrete Problem

Describe photon showers in high granularity calorimeter prototype

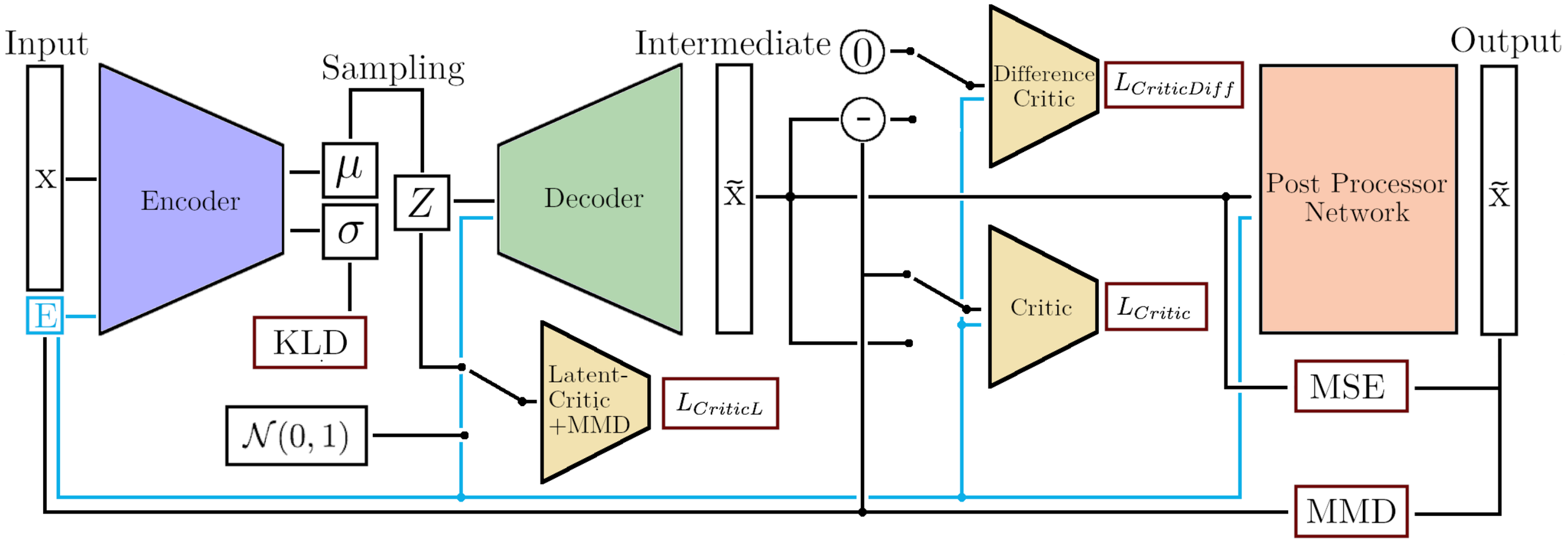
- 30x30x30 cells (Si-W)
- Photon energies from 10 to 100 GeV
- Use 950k examples (uniform in energy) created with GEANT4 to train





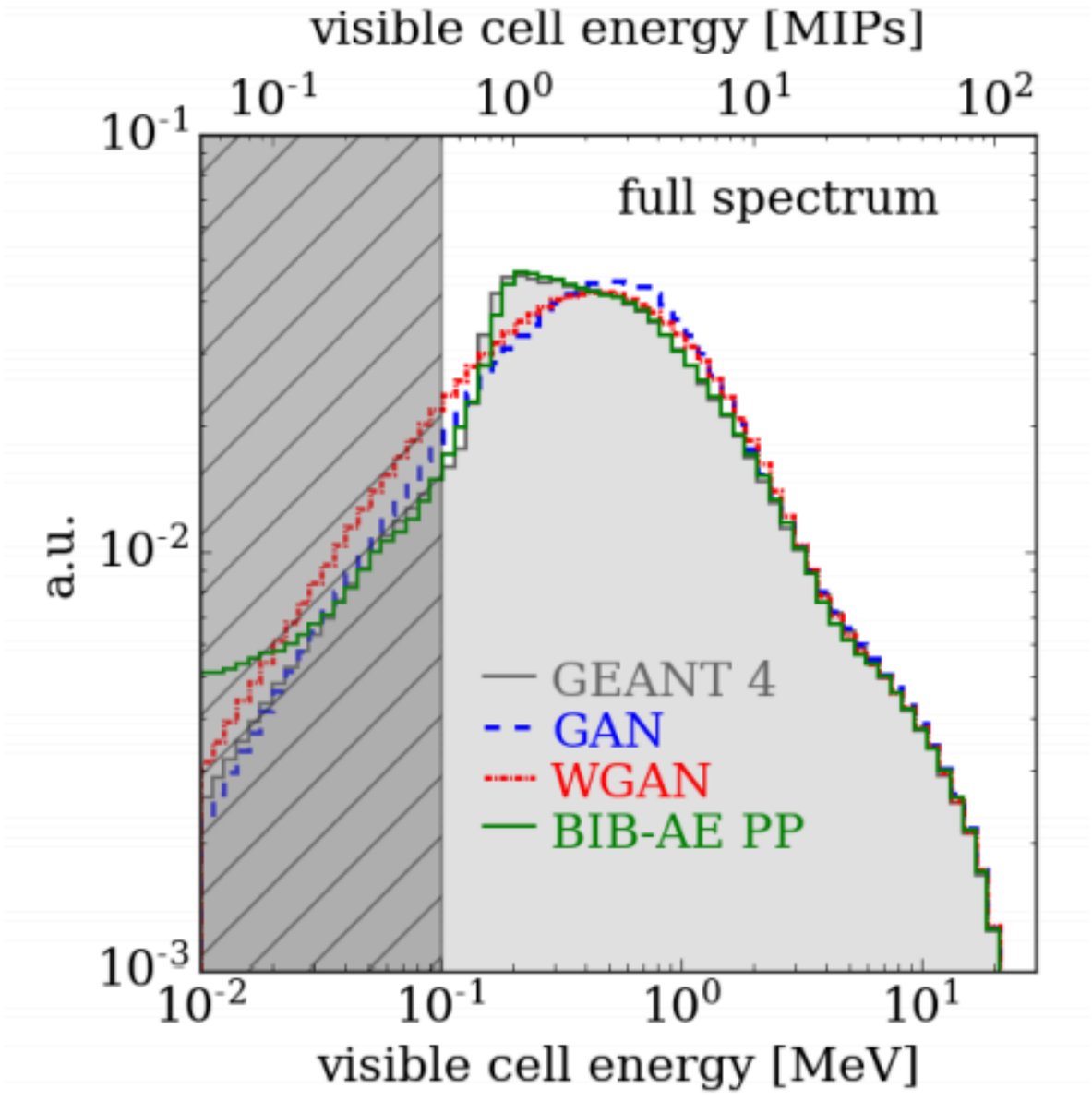
- Not only model individual images but also **differential distributions**

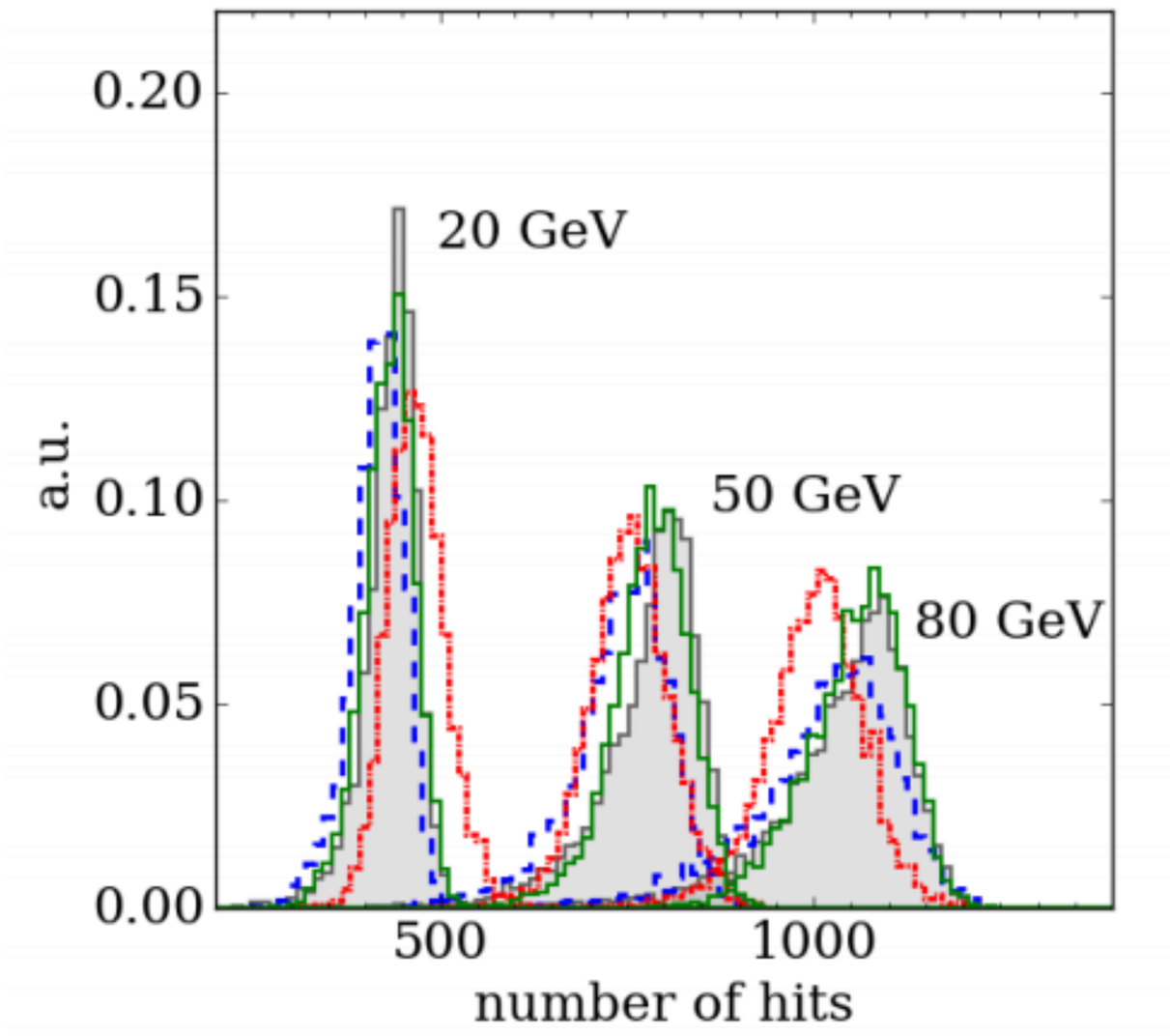
Architecture



- BIB-AE (based on 1912.00830) with added post-processing
- Unifies features of GAN and VAE
- 71M trainable parameters

Result



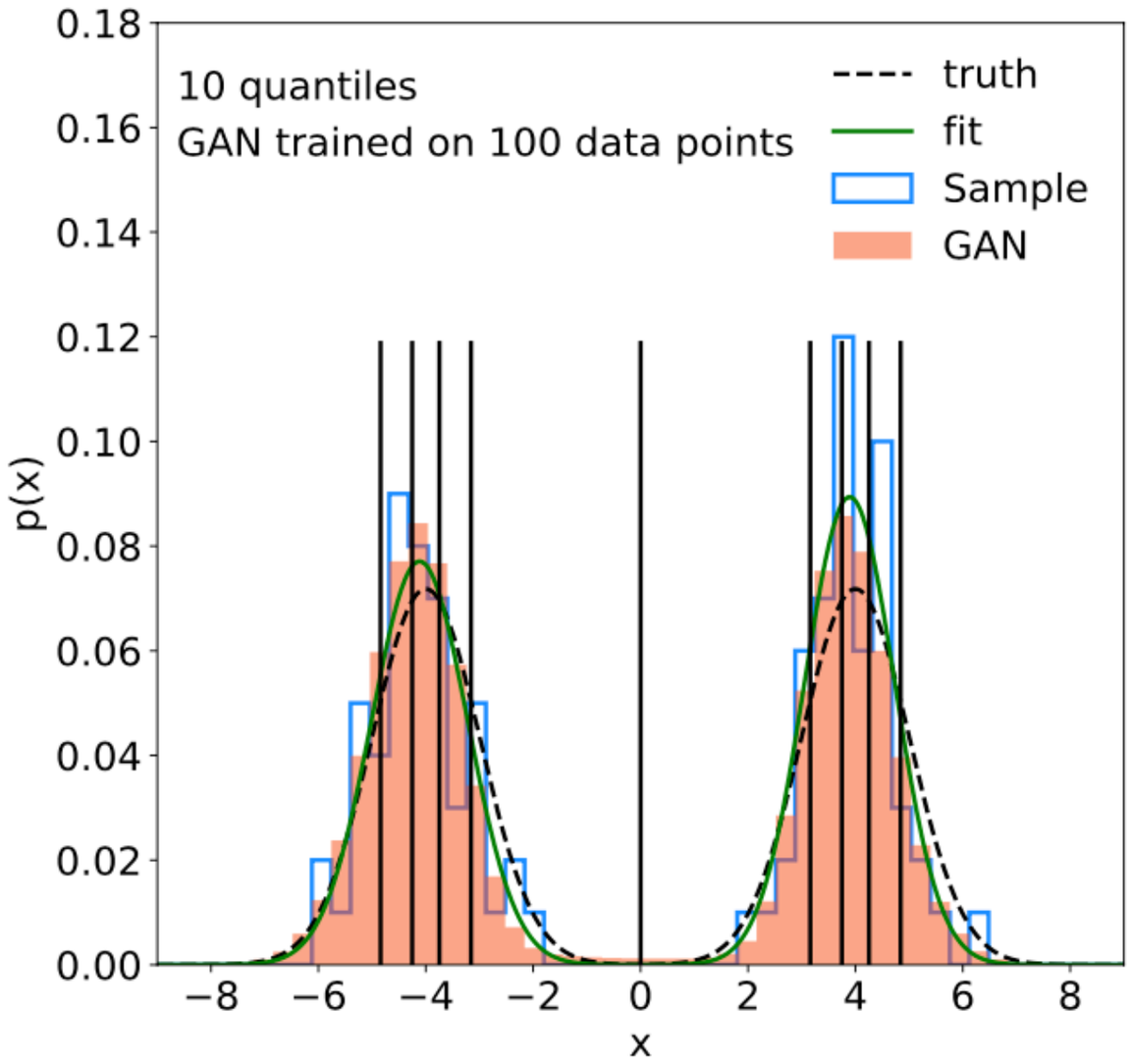


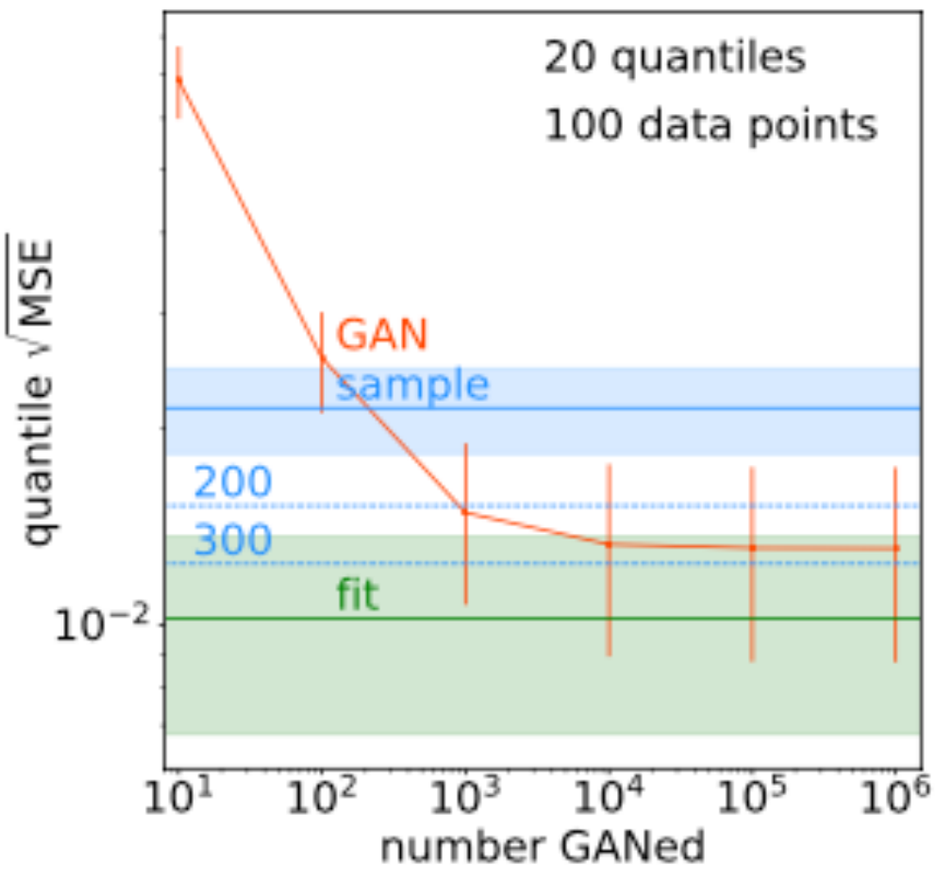
Can now learn differential distributions
Still room to improve

Limitations of Generative Models

- Generative models are powerful in quickly producing more examples, still need training examples
- Machine learning is great at interpolation, but it cannot do magic
- Expect to simulate typical examples, do not trust the tails of distributions without verification
- Can networks **amplify?**

Amplification 1D

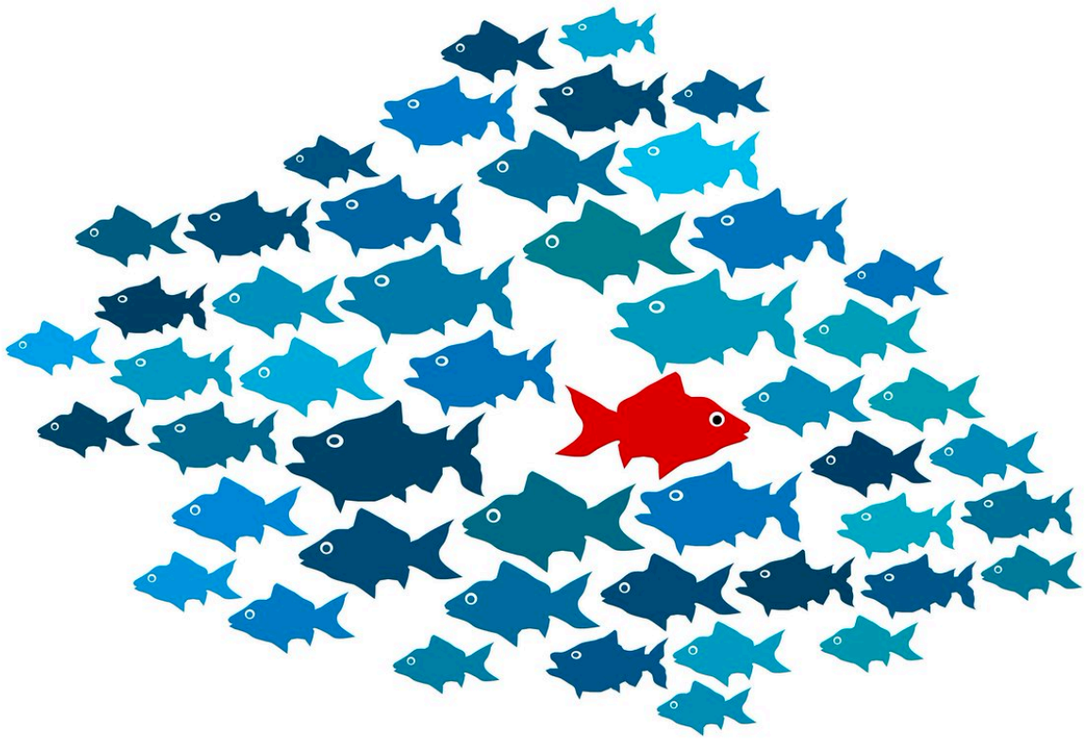




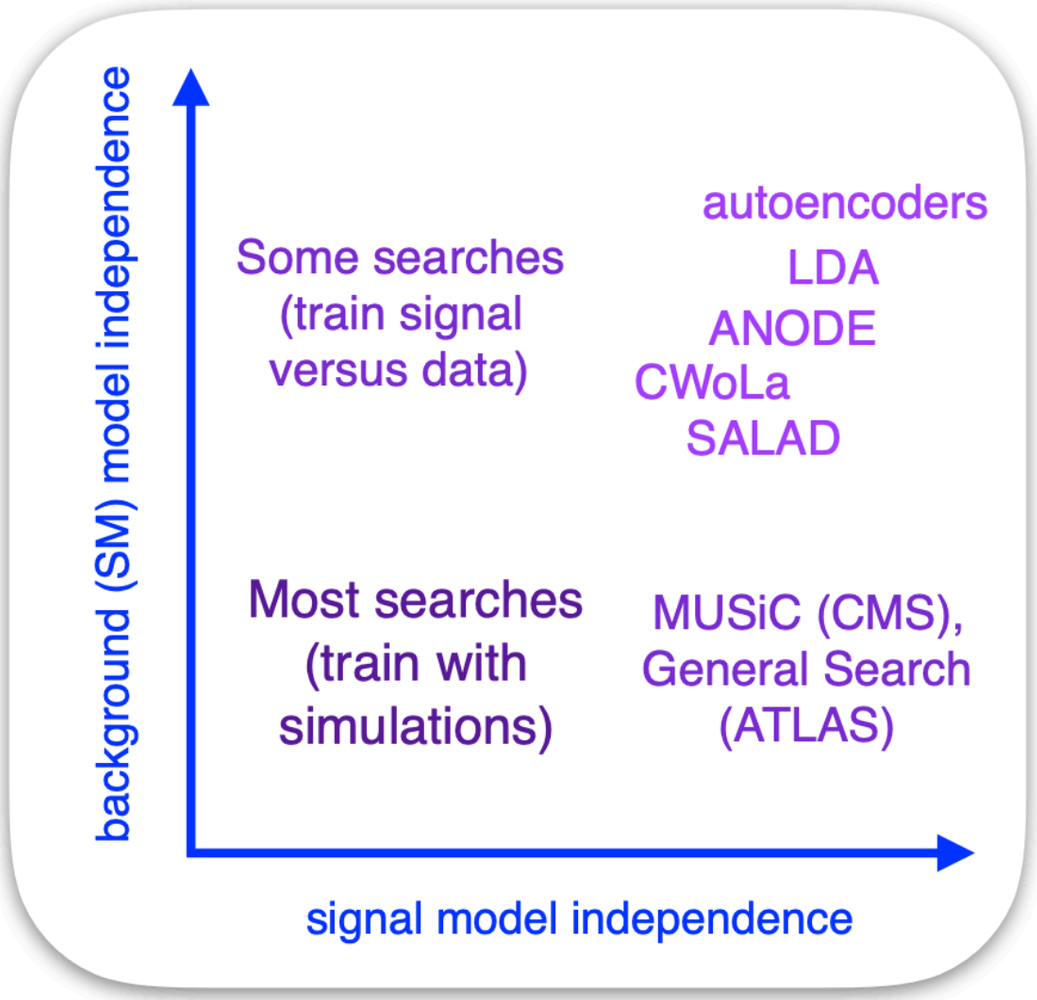
Improve statistics of training sample by interpolation

Unsupervised Searches

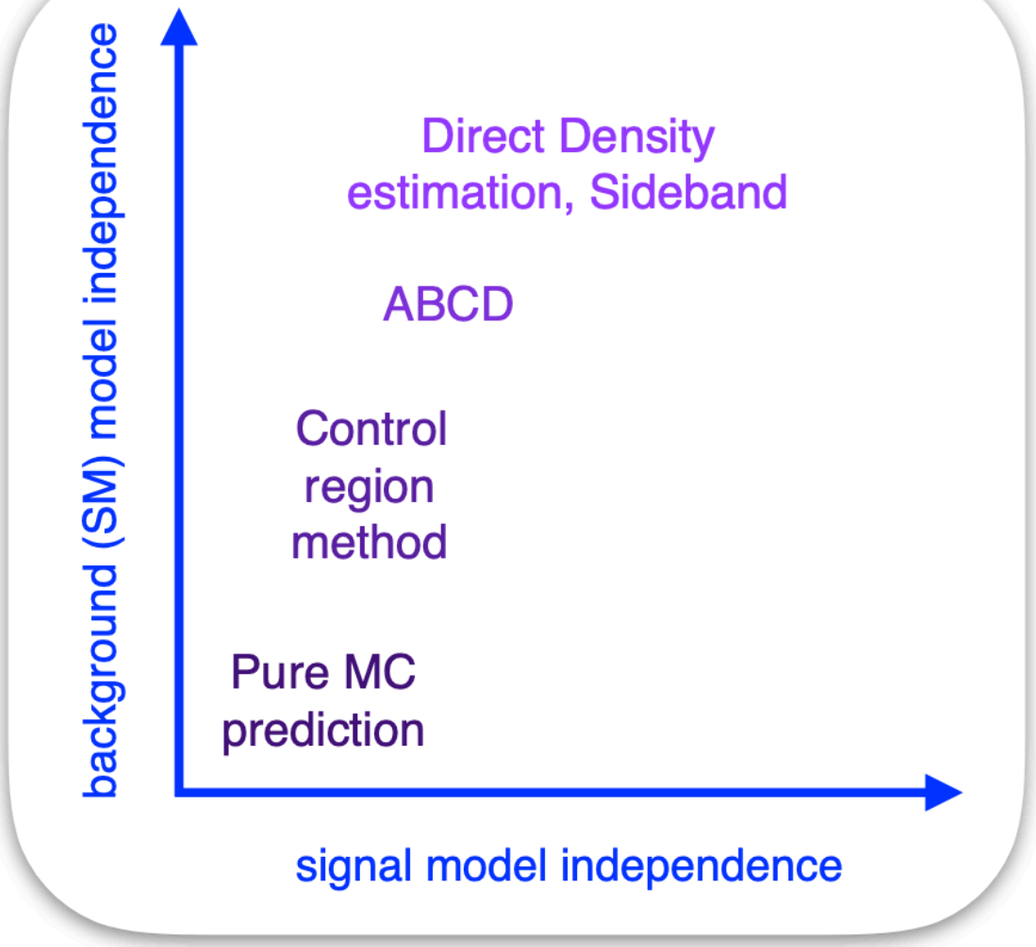
Can we look for new physics, without knowing what to look for?



Approaches



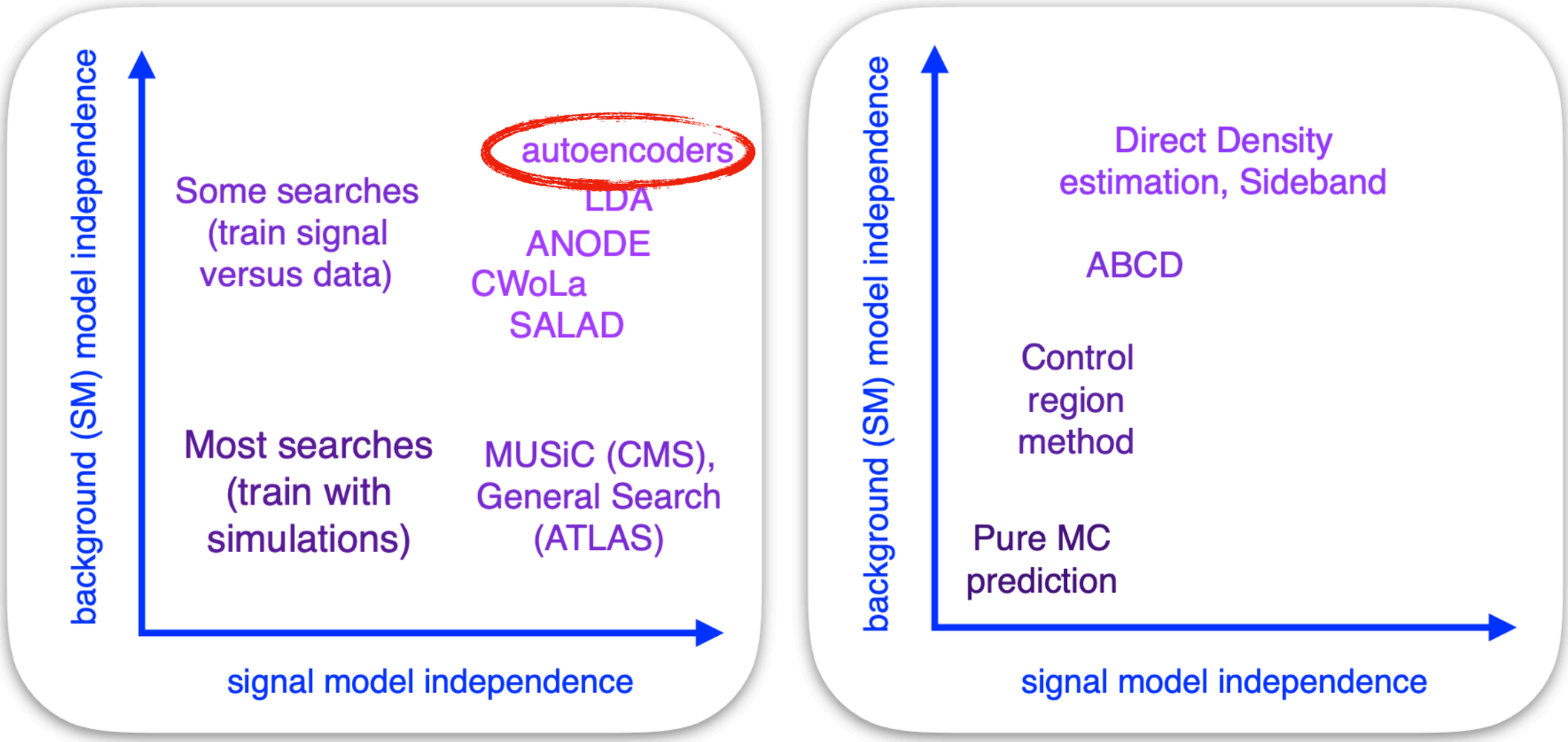
(a) Signal sensitivity



(b) Background specificity

From Ben Nachman, David Shih, 2001.04990

Approaches

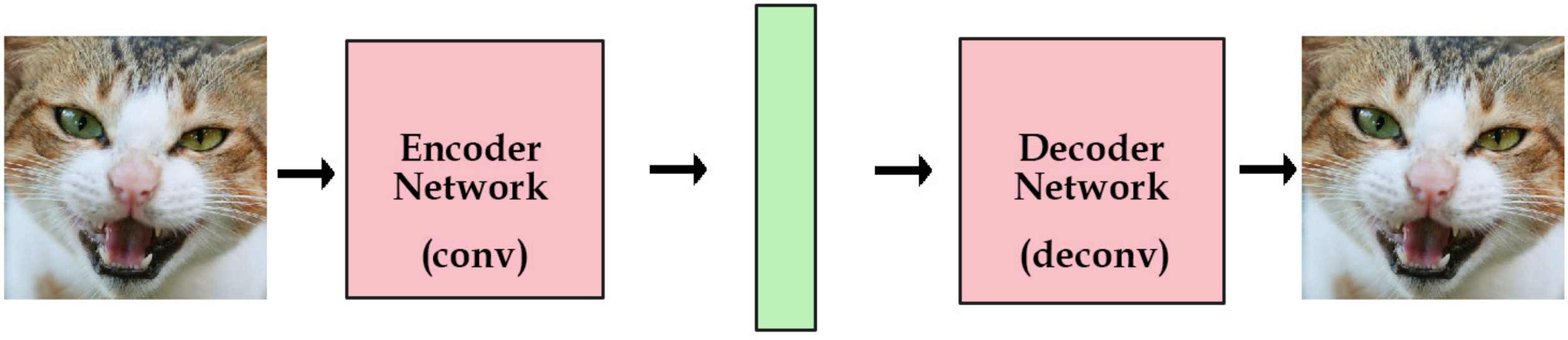


(a) Signal sensitivity

(b) Background specificity

From Ben Nachman, David Shih, 2001.04990

Autoencoder

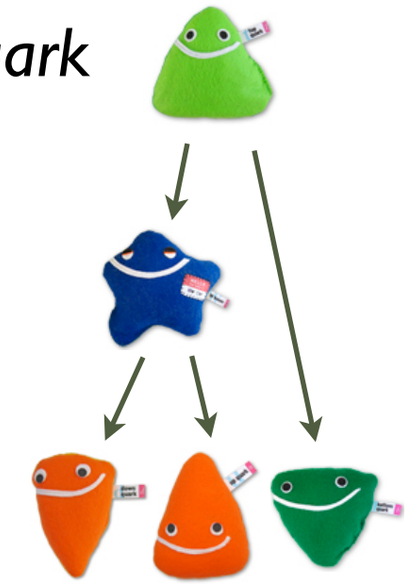


$f(x)$ **latent vector / variables** $g(f(x))$

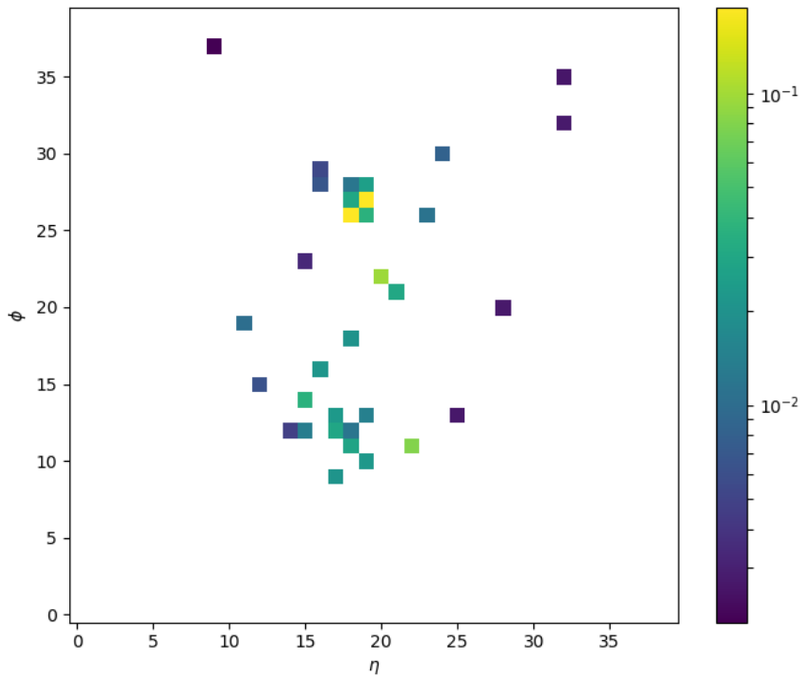
$$L = (\hat{y} - g(f(x)))^2$$

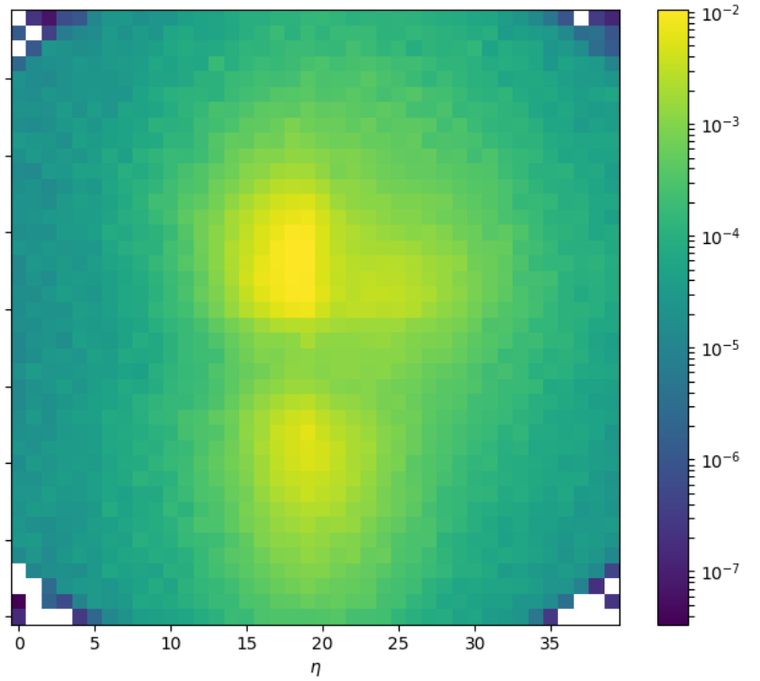
- Weakly supervised learning
- *Latent space/bottleneck* with compressed representation
- Dimension reduction
- Denoising

Top Quark Jet



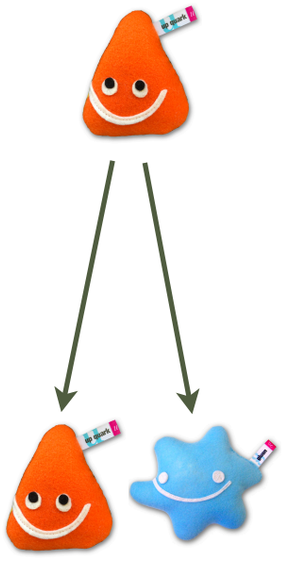
=



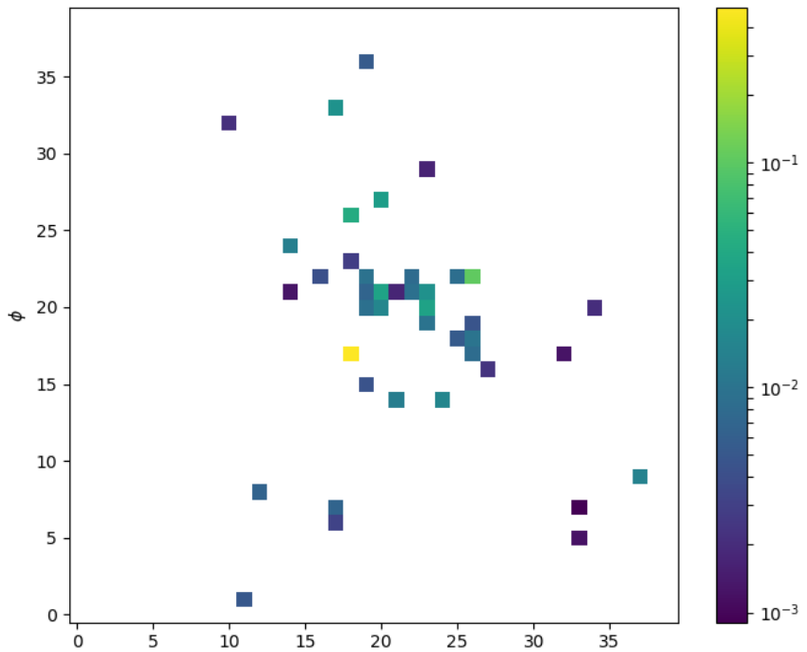


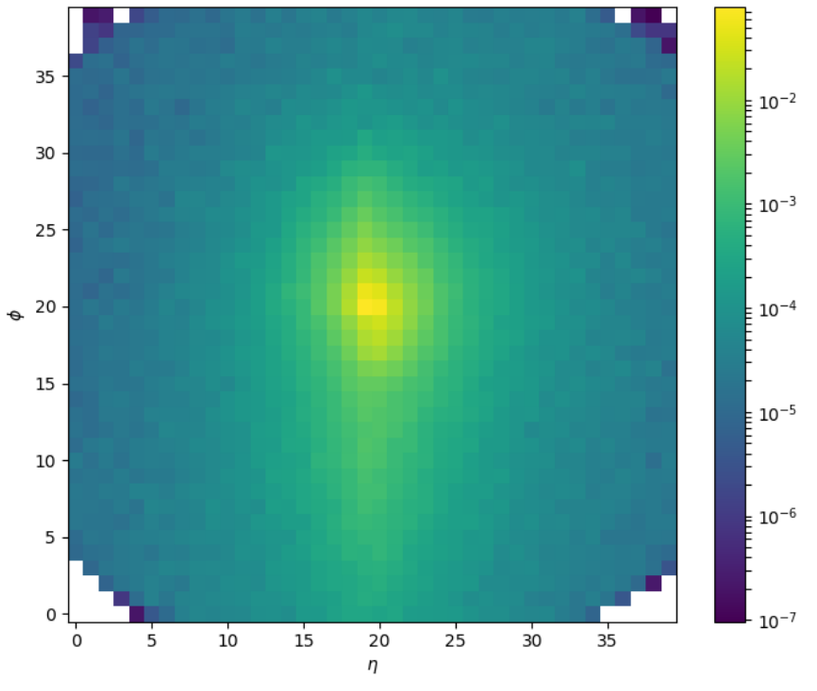
Remember Jet Images

QCD Jet

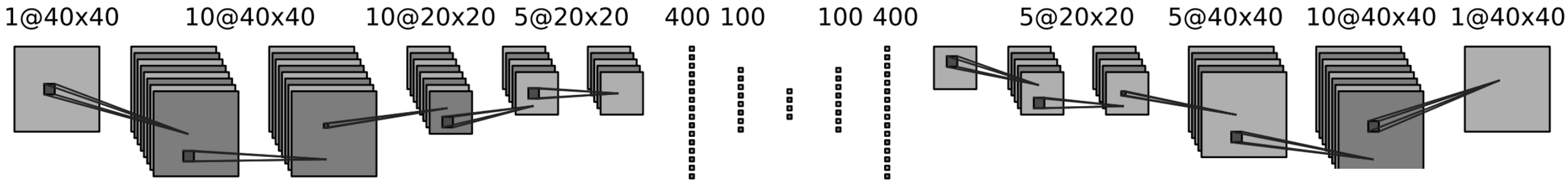


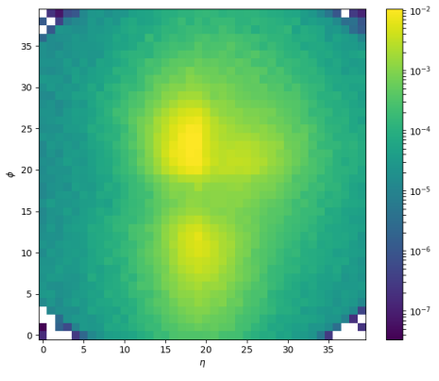
=



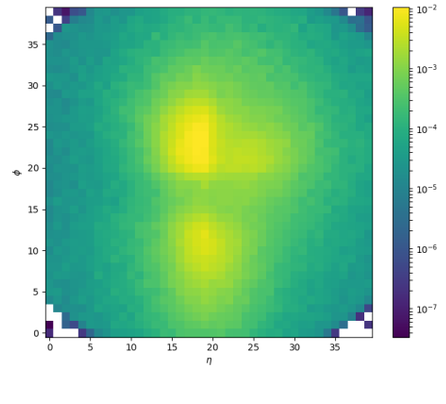


Autoencoder

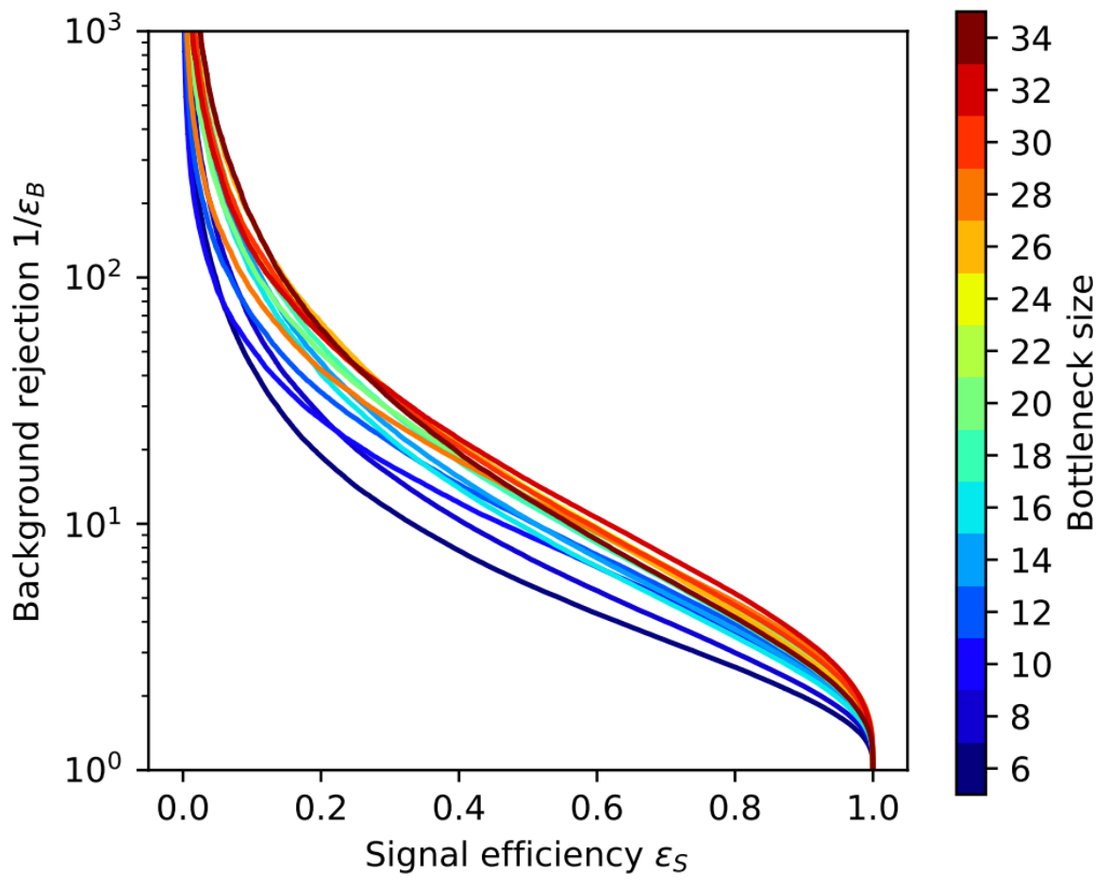




$$L_{\text{auto}} = \sum_{\text{1600 pixels}} \left(k_T^{\text{norm,in}} - k_T^{\text{auto}} \right)^2$$



- Train on pure QCD light quark/ gluon jets and apply to top tagging
- Top quarks/ new physics identified as anomaly



QCD or What?
T Heimel, GK, T Plehn, JM Thompson, 1808.08979
Searching for New Physics with Deep Autoencoders
M Farina, Y Nakai, D Shih, 1808.08992

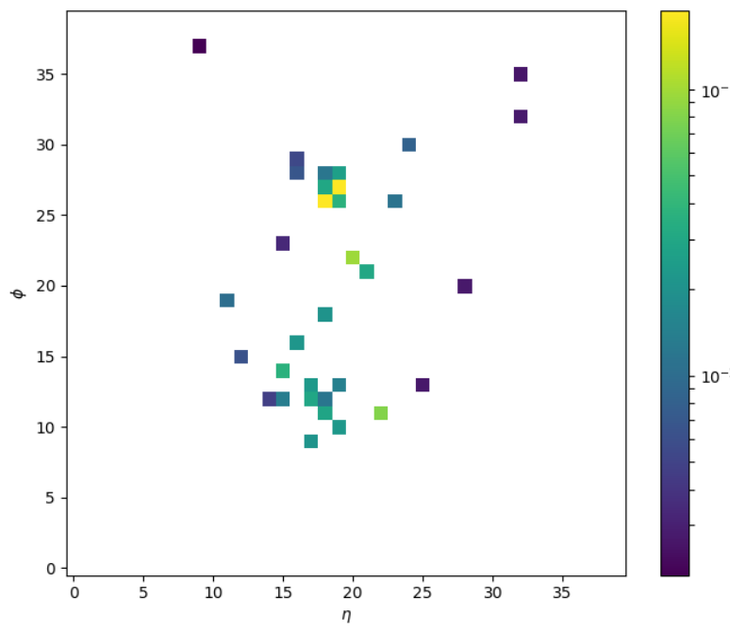
Caveats

- Anomaly score for a given signature depends on complexity of signal/background in addition to training data

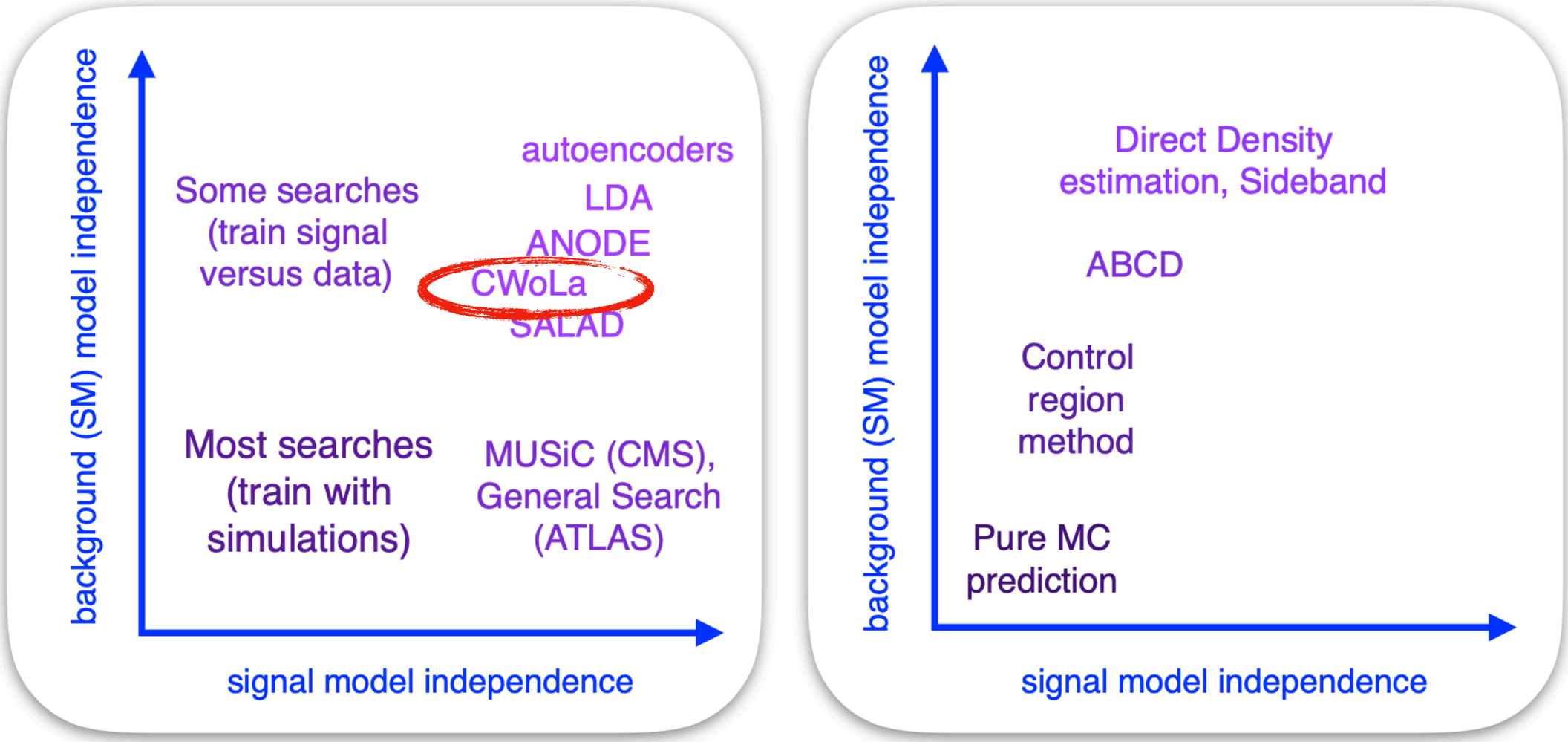
- We are not looking for individual anomalous events but anomalous regions of phase space

- Usual L2 difference not optimal as loss:

  - Different distributions of pixels compatible with same physics



Approaches

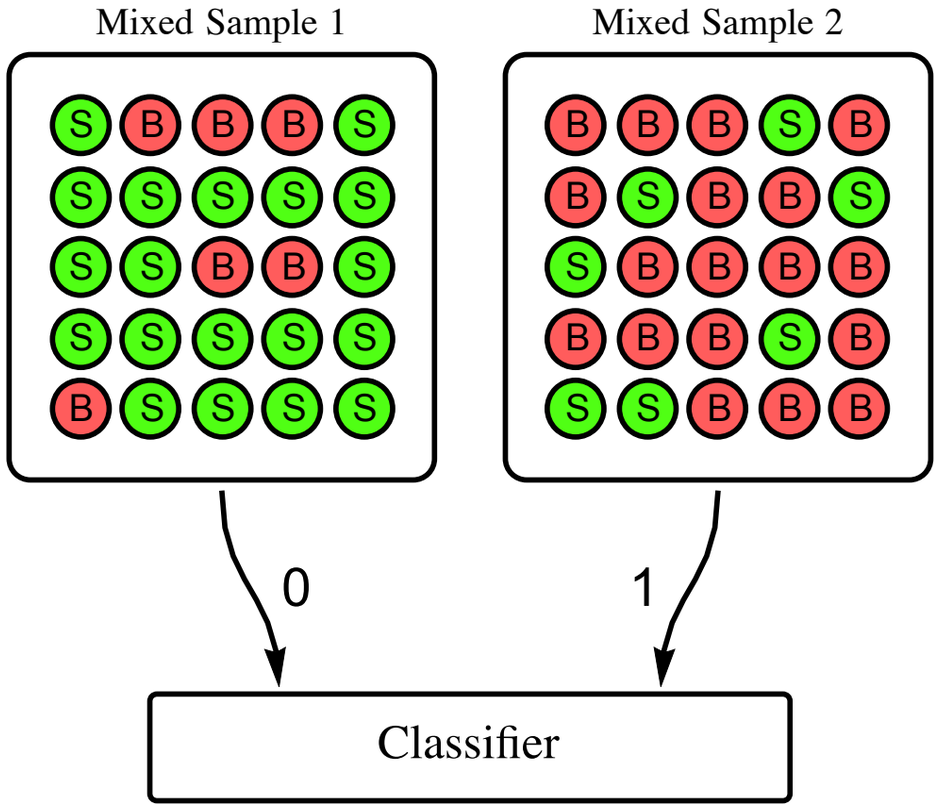


(a) Signal sensitivity

(b) Background specificity

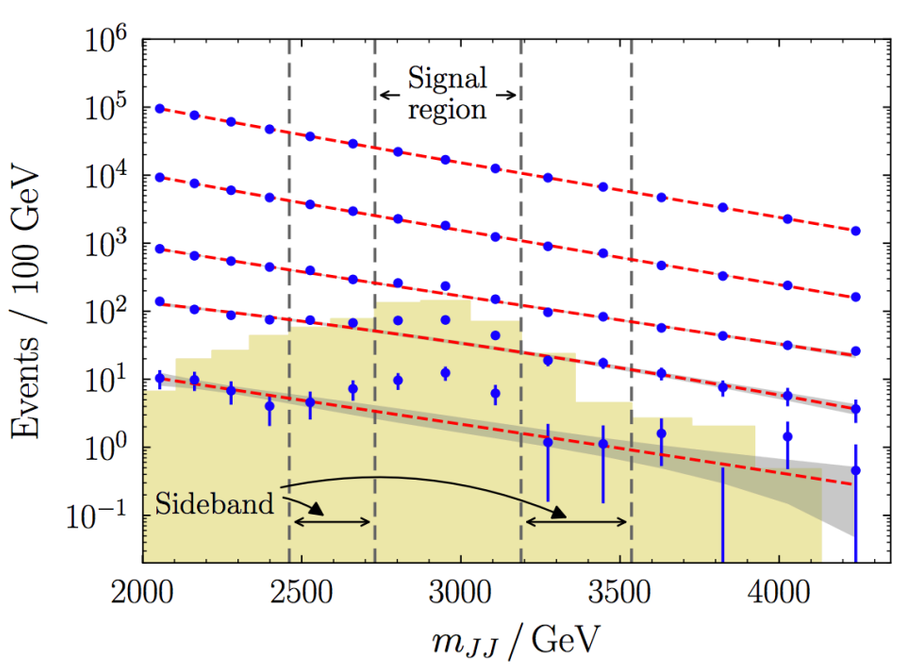
From Ben Nachman, David Shih, 2001.04990

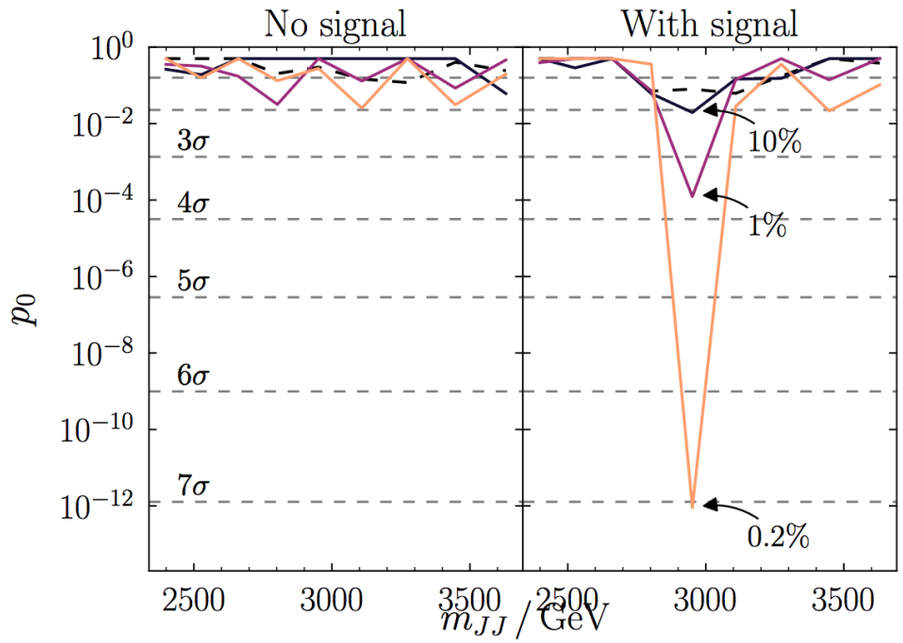
CWola Hunting



$$L_{M_1/M_2} = \frac{p_{M_1}}{p_{M_2}} = \frac{f_1\, p_S + (1-f_1)\, p_B}{f_2\, p_S + (1-f_2)\, p_B} = \frac{f_1\, L_{S/B} + (1-f_1)}{f_2\, L_{S/B} + (1-f_2)}$$

Distinguishing mixed samples is equivalent to signal/background classification!

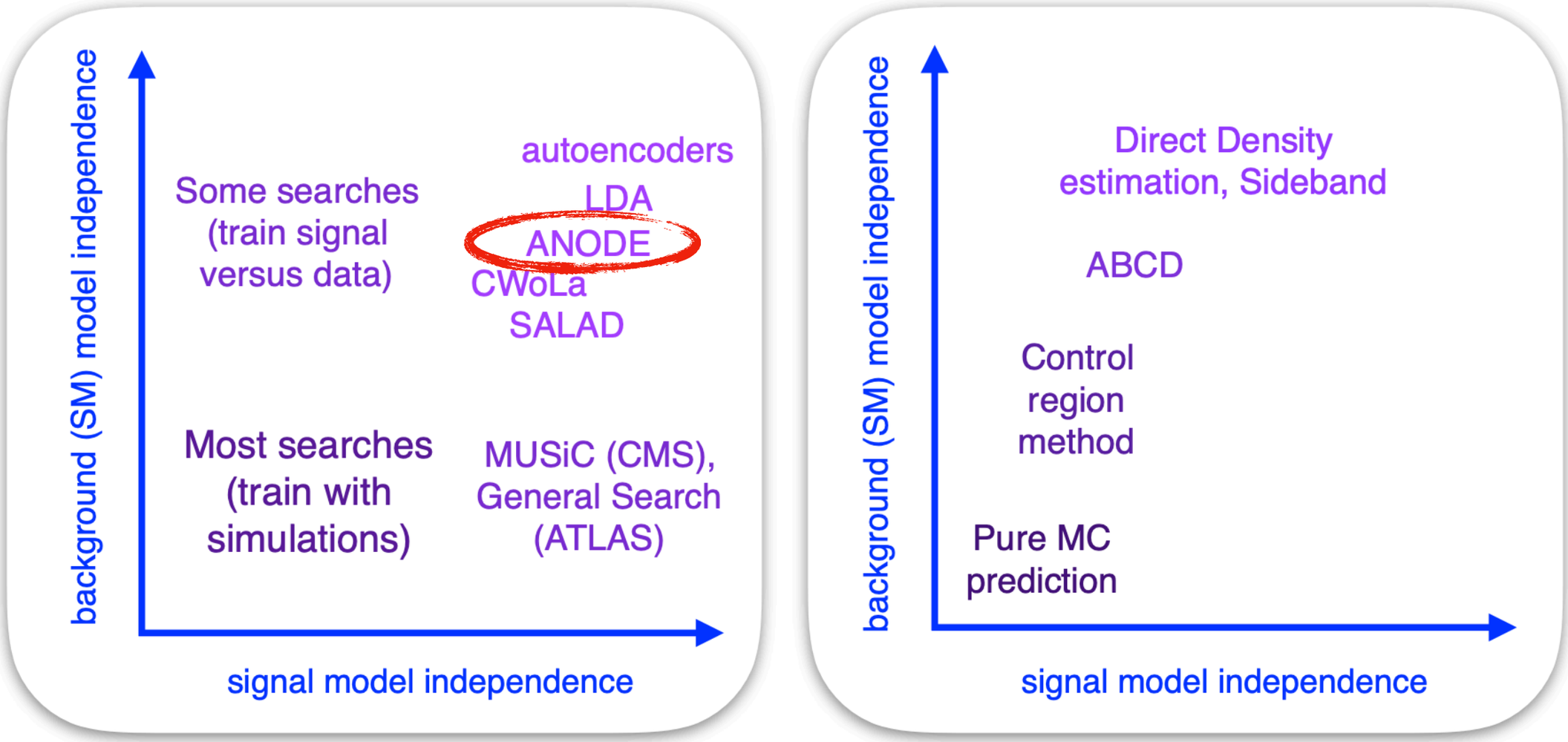




- Assume signal is resonant in one variable
- Define signal region and sidebands
- Train classifier and look for excess

Classification without labels: Learning from mixed samples in high energy physics, EM Metodiev, B Nachman, J Thaler, 1708.02949
Anomaly Detection for Resonant New Physics with Machine Learning JH Collins, K Howe, B Nachman 1805.02664

Approaches



(a) Signal sensitivity

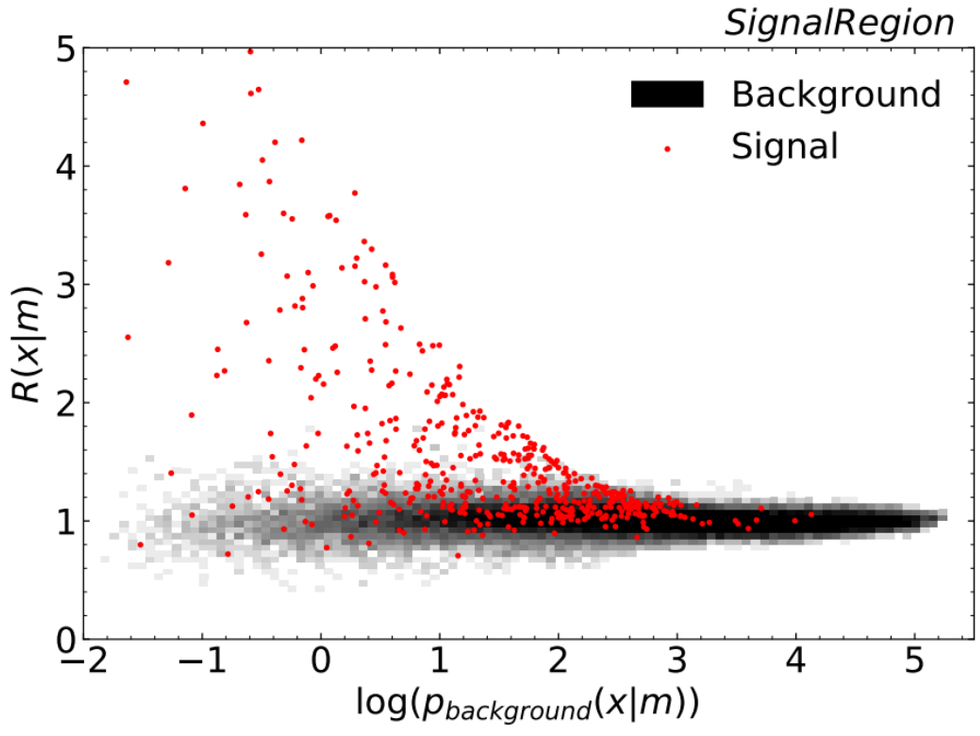
(b) Background specificity

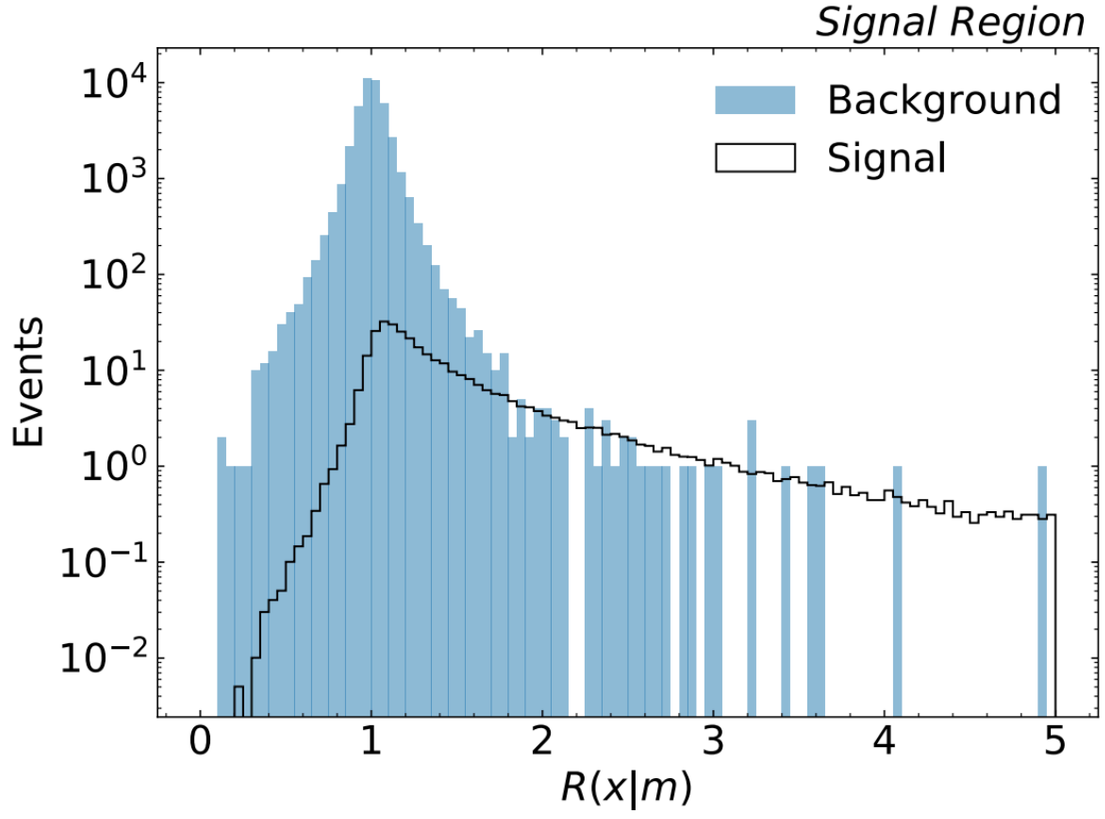
From Ben Nachman, David Shih, 2001.04990

ANODE: ANOmaly detection with Density Estimation

An anomaly is a local over density of events

- Build density estimator in sideband region P_{SB}
- Extrapolation to signal region gives background estimate P_{SB} -> P_{BG}
- Build density estimator in signal region P_{SR}
- Likelihood ratio $R=P_{SR}/P_{BG}$
- *Density estimation via MAF (1705.07057) (Masked Autoregressive Flow)*





Anomaly Detection with Density Estimation, B Nachman, D Shhih 2001.04990

Other Ideas

- Naively Autoencoder more sensitive to outliers (out-of-data examples), density estimation more sensitive to anomalies in distributions
- One could also look for density anomalies in the latent space of autoencoders
- Also very interesting for non-HEP applications:
  - Data quality monitoring
  - Predictive maintenance
  - Credit card fraud
  - …..
- Exciting topic to start now!

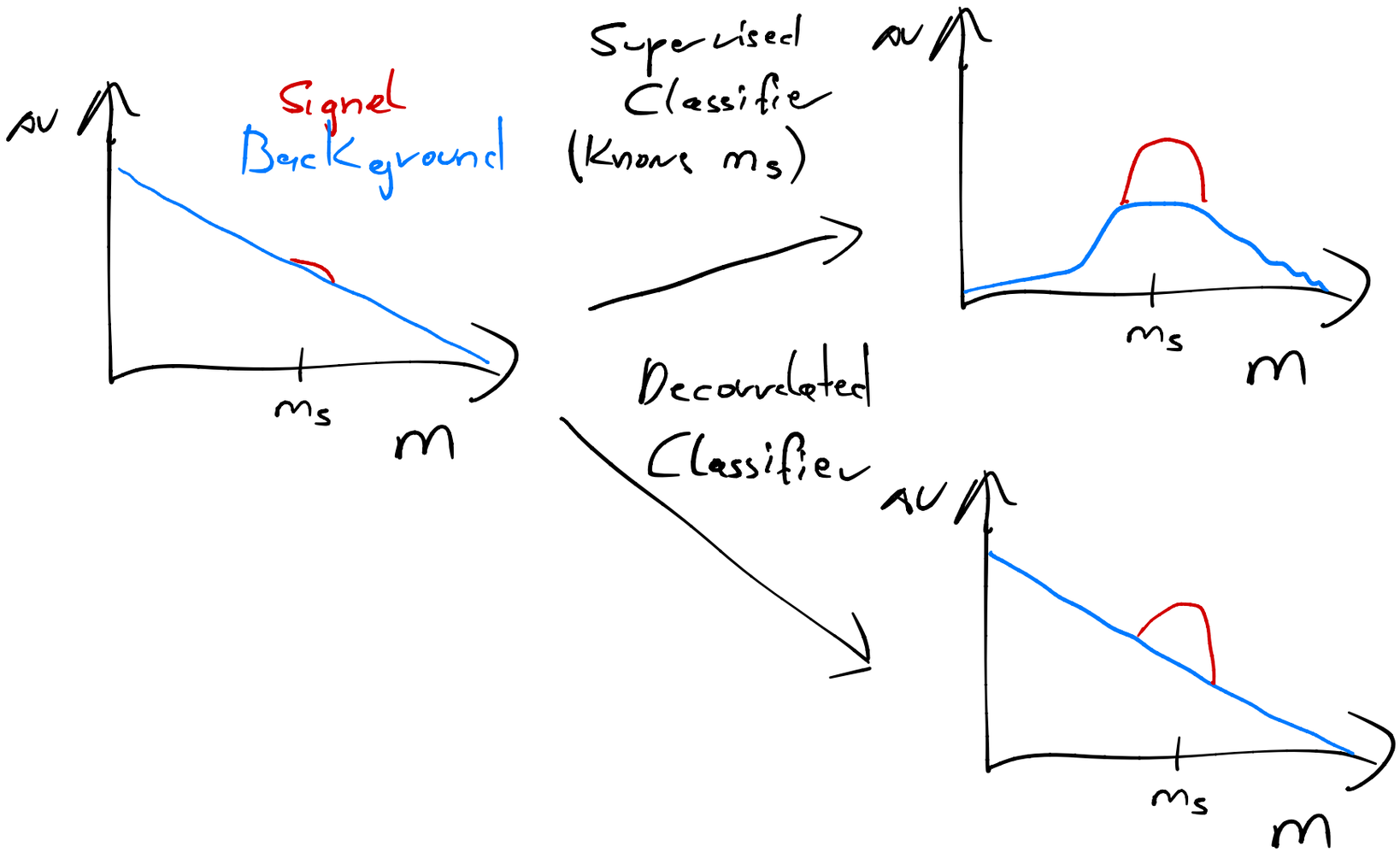
LHC Olympics 2020

- For more on anomaly detection see material at the recent workshop: https://indico.desy.de/e/anomaly2020

Some final words

Correlation

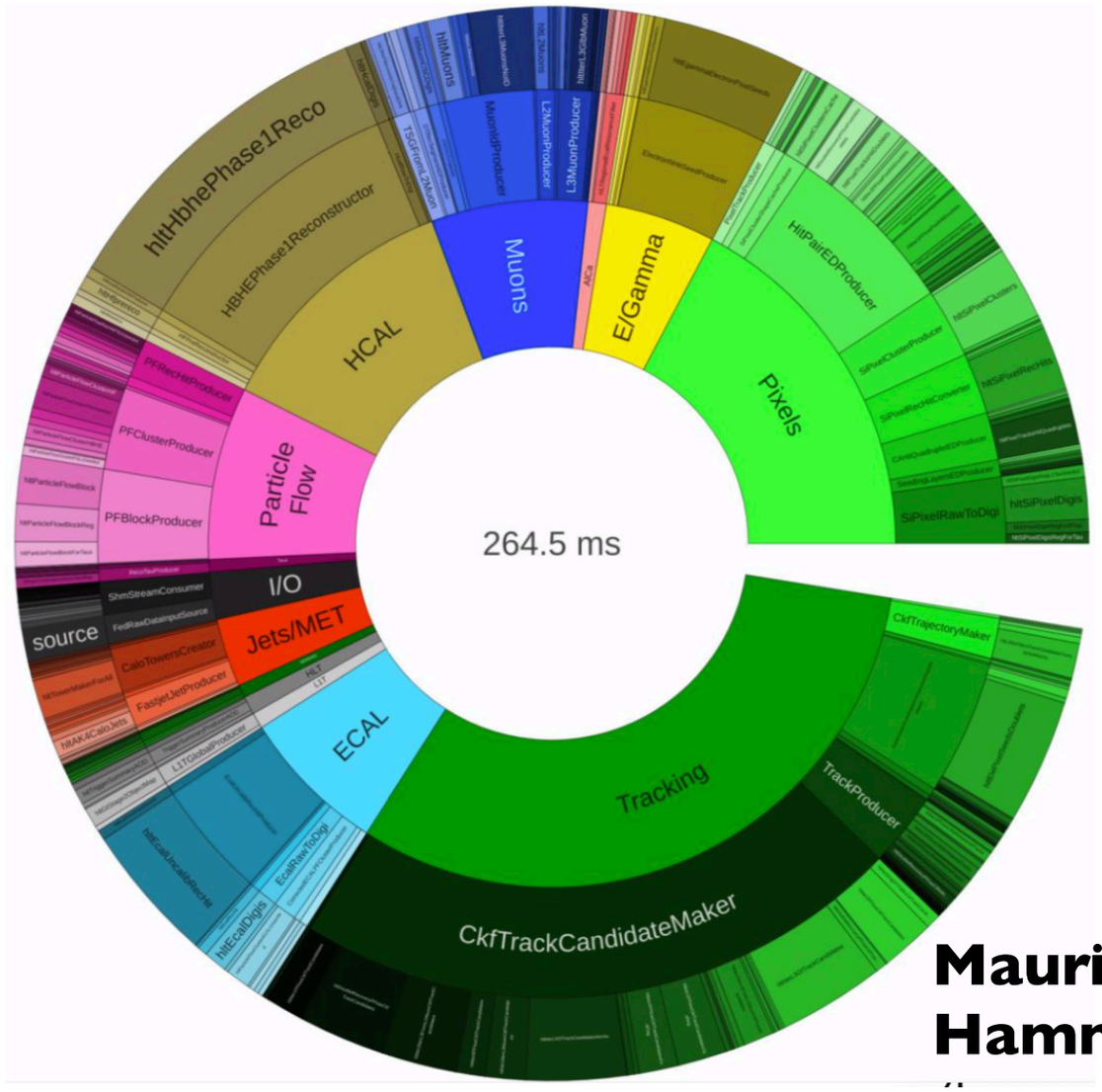
Against a variable or data vs simulation

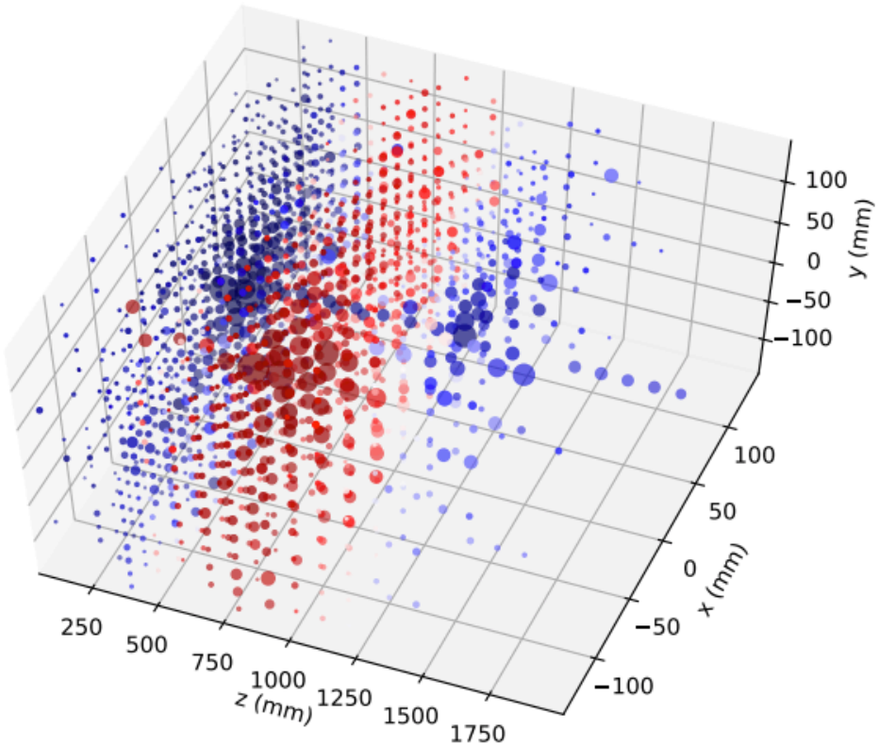


- Large number of ideas including planing (1908.08959), adversarial training (1611.01046,1703.03507), DisCo (2001.05310),…

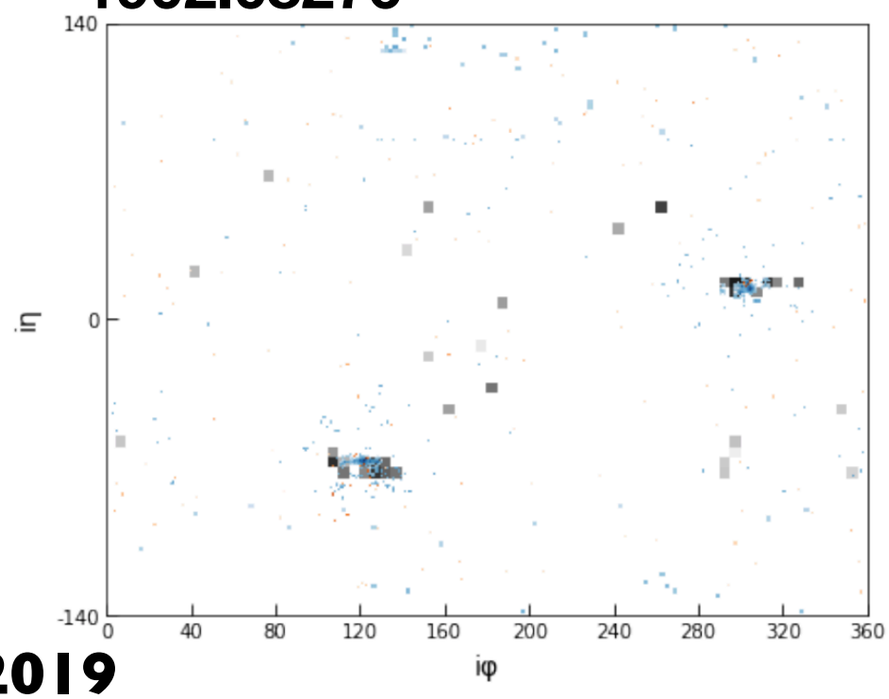
Low Level Reconstruction

- Replace traditional algorithms for reconstruction, object ID and calibration with deep learning
- Increase physics performance and/or resource usage
- Superficially less attractive, potentially much more useful
- End-to-end learning?



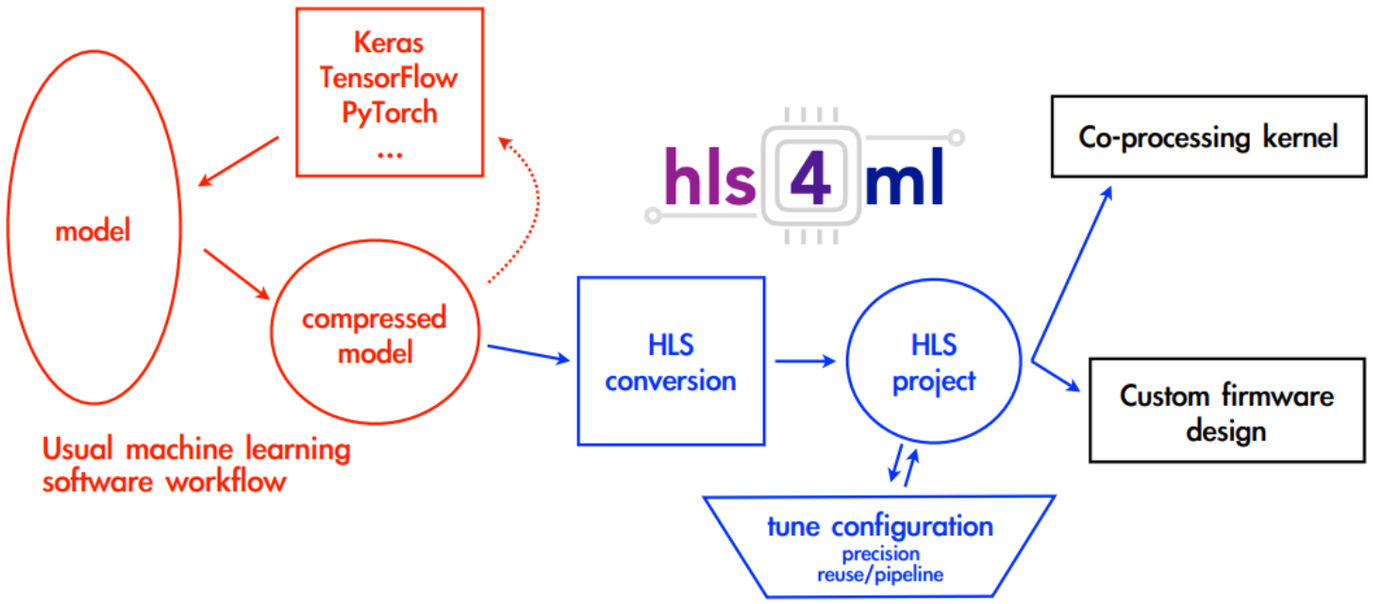


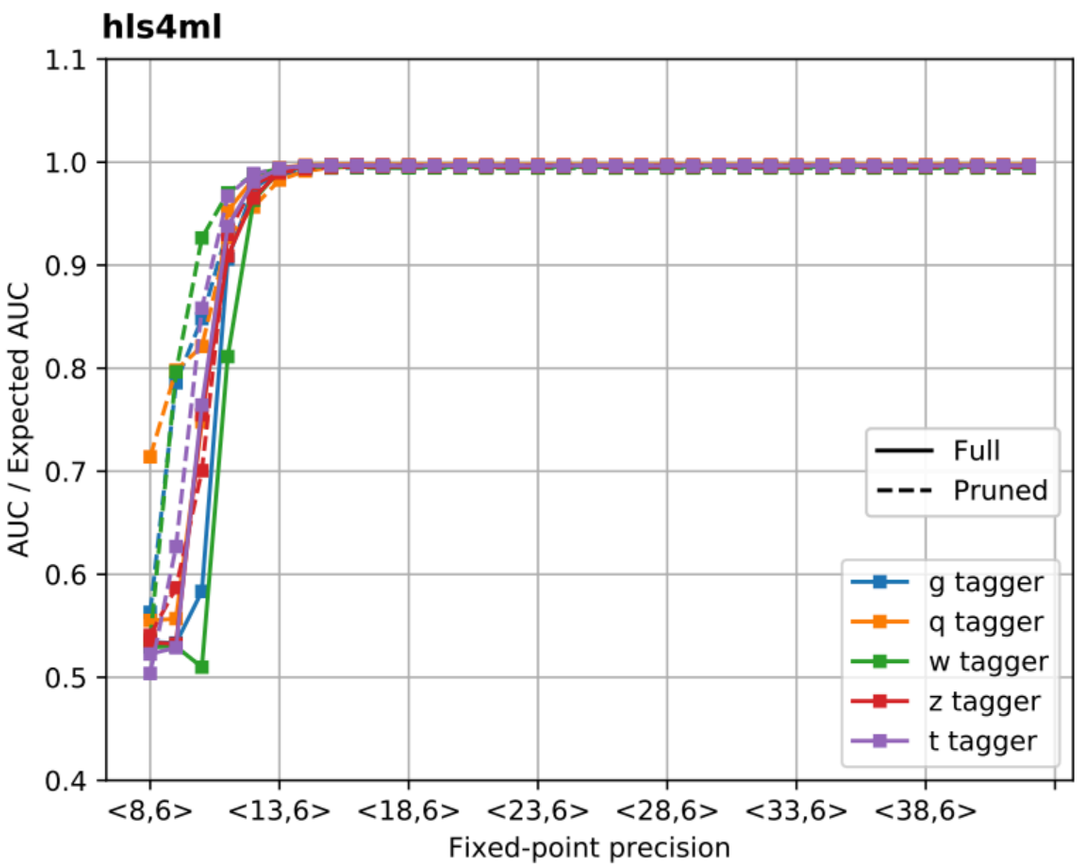
1902.08276

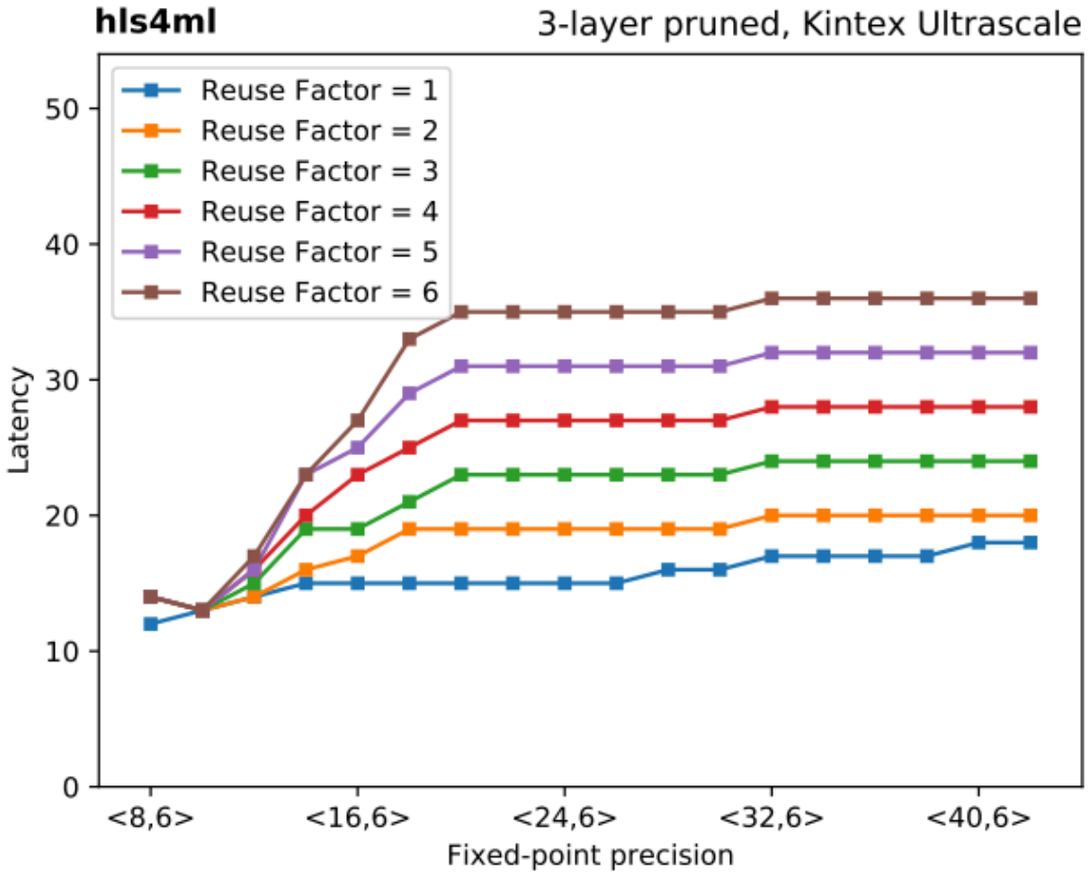


Fast Decisions

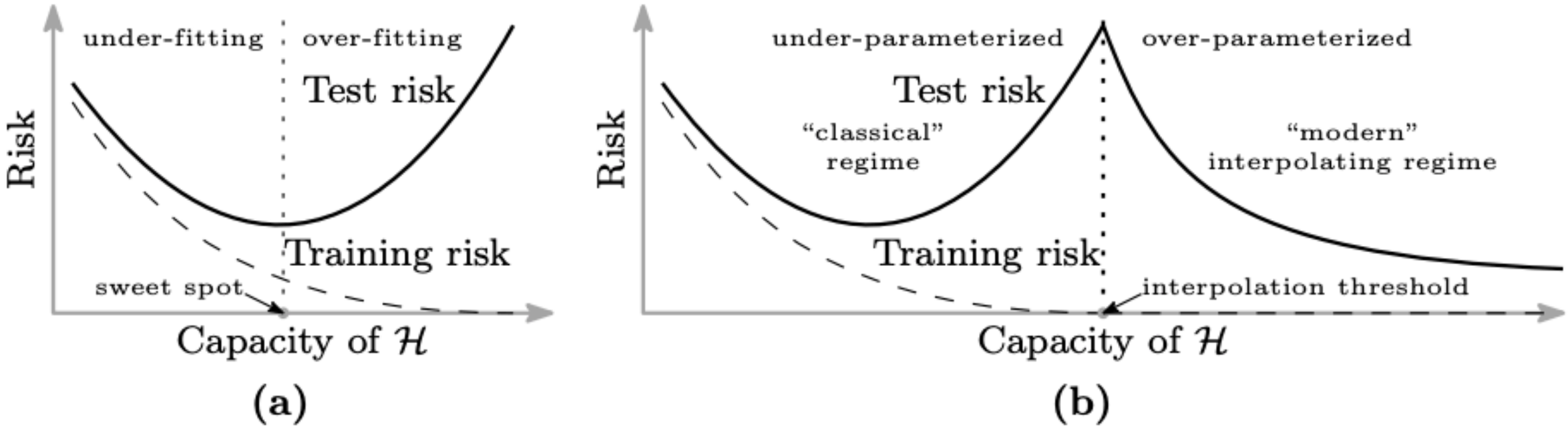
- Use neural networks in L1 Trigger
- Trained offline using normal tools, then translated and optimised for running on FPGAs







Overtraining



1812.11118

Conclusions

- Deep Learning for particle physics is rapidly developing solutions to a wide range of problems
  - Object and Event classification
  - Anomaly detection
  - Robustness and uncertainties
  - Fast reconstruction and simulation
- Further reading
  - Basic concepts: http://www.deeplearningbook.org/
  - Overview of ML in HEP papers: https://iml-wg.github.io/HEPML-LivingReview/

Thank you!